EMNLP 2020

Fifth Conference on Machine Translation

Proceedings of the Conference

November 19-20, 2020
Online

Order copies of this and other ACL proceedings from:

Association for Computational Linguistics (ACL)
209 N. Eighth Street
Stroudsburg, PA 18360
USA
Tel: +1-570-476-8006
Fax: +1-570-476-0860
acl@aclweb.org

ISBN 978-1-948087-81-0

Introduction

The Fifth Conference on Machine Translation (WMT 2020) took place on Thursday, November 19 and Friday, November 20, 2020 immediately following the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020).

This is the fifth time WMT has been held as a conference. The first time WMT was held as a conference was at ACL 2016 in Berlin, Germany, the second time at EMNLP 2017 in Copenhagen, Denmark, the third time at EMNLP 2018 in Brussels, Belgium, and the fourth time at ACL 2019 in Florence, Italy. Prior to being a conference, WMT was held 10 times as a workshop. WMT was held for the first time at HLT-NAACL 2006 in New York City, USA. In the following years the Workshop on Statistical Machine Translation was held at ACL 2007 in Prague, Czech Republic, ACL 2008, Columbus, Ohio, USA, EACL 2009 in Athens, Greece, ACL 2010 in Uppsala, Sweden, EMNLP 2011 in Edinburgh, Scotland, NAACL 2012 in Montreal, Canada, ACL 2013 in Sofia, Bulgaria, ACL 2014 in Baltimore, USA, EMNLP 2015 in Lisbon, Portugal.

The focus of our conference is to bring together researchers from the area of machine translation and invite selected research papers to be presented at the conference.

Prior to the conference, in addition to soliciting relevant papers for review and possible presentation, we conducted 11 shared tasks. These consisted of seven translation tasks: Machine Translation of News, Lifelong Learning for Machine Translation, Robust Machine Translation, Similar Language Translation, Unsupervised and Very Low Resource Supervised Translation, Biomedical Translation, and Machine Translation for Chats, and four other tasks: Automatic Post-Editing, Metrics for Machine Translation, and Parallel Corpus Filtering and Alignment for Low-Resource Conditions.

The results of all shared tasks were announced at the conference, and these proceedings also include overview papers for the shared tasks, summarizing the results, as well as providing information about the data used and any procedures that were followed in conducting or scoring the tasks. In addition, there are short papers from each participating team that describe their underlying system in greater detail.

Like in previous years, we have received a far larger number of submissions than we could accept for presentation. WMT 2020 has received 58 full research paper submissions (not counting withdrawn submissions). In total, WMT 2020 featured 19 full research paper oral presentations and 112 shared task poster presentations.

The invited talk entitled "Low-resourcedness Beyond Data" was given by Ignatius Ezeani, Jade Abbott, Julia Kreutzer, Salomon Kabongo, Perez Ogayo, Shamsuddeen Hassan Muhammad, Rubungo Andre Niyongabo, Jamiil Toure Ali, Kathleen Siminyu, Salomey Osei, Wilhelmina Nekoto, Arshath Ramkilowan, Masabata Mokgesi-Selinga, Bonaventure Dossou, Ayodele Olabiyi, Blessing Sibanda, Akinola Oluwole, Vukosi Marivate, and Orevaoghene Ahia.

We would like to thank the members of the Program Committee for their timely reviews. We also would like to thank the participants of the shared task and all the other volunteers who helped with the evaluations.

Loïc Barrault, Ondřej Bojar, Fethi Bougares, Rajen Chatterjee, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Alexander Fraser, Yvette Graham, Paco Guzman, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, André Martins, Makoto Morishita, Christof Monz,

Masaaki Nagata, Toshiaki Nakazawa, Matteo Negri, Aurélie Névéol, Mariana Neves, Martin Popel, Matt Post, Marco Turchi, Marcos Zampieri.

Co-Organizers

Organizers:

Loïc Barrault (University of Sheffield)
Ondřej Bojar (Charles University in Prague)
Fethi Bougares (University of Le Mans)
Rajen Chatterjee (Apple)
Marta R. Costa-jussà (Universitat Politècnica de Catalunya)
Christian Federmann (MSR)
Mark Fishel (University of Tartu)
Alexander Fraser (LMU Munich)
Yvette Graham (DCU)
Paco Guzman (Facebook)
Barry Haddow (University of Edinburgh)
Matthias Huck (LMU Munich)
Antonio Jimeno Yepes (IBM Research Australia)
Philipp Koehn (Johns Hopkins University)
André Martins (Unbabel)
Makoto Morishita (NTT)
Christof Monz (University of Amsterdam)
Masaaki Nagata (NTT)
Toshiaki Nakazawa (University of Tokyo)
Matteo Negri (FBK)
Aurélie Névéol (LIMSI, CNRS)
Mariana Neves (German Federal Institute for Risk Assessment)
Martin Popel (Charles University in Prague)
Matt Post (Johns Hopkins University)
Marco Turchi (FBK)
Marcos Zampieri (Rochester Institute of Technology)

Invited Speakers:

Ignatius Ezeani, Jade Abbott, Julia Kreutzer, Salomon Kabongo, Perez Ogayo, Shamsuddeen Hassan Muhammad, Rubungo Andre Niyongabo, Jamiil Toure Ali, Kathleen Siminyu, Salomey Osei, Wilhelmina Nekoto, Arshath Ramkilowan, Masabata Mokgesi-Selinga, Bonaventure Dossou, Ayodele Olabiyi, Blessing Sibanda, Akinola Oluwole, Vukosi Marivate, and Orevaoghene Ahia

Program Committee:

Tamer Alkhouli (AppTek)
Antonios Anastasopoulos (George Mason University)
Yuki Arase (Osaka University)
Mihael Arcan (National Universith of Ireland Galway)
Philip Arthur (Monash University)
Duygu Ataman (University of Zürich)

Eleftherios Avramidis (German Research Center for Artificial Intelligence (DFKI))
Amittai Axelrod (DiDi Labs)
Parnia Bahar (RWTH Aachen University)
Rachel Bawden (University of Edinburgh)
Meriem Beloucif (University of Hamburg)
Chris Brockett (Microsoft Research)
Ozan Caglayan (Imperial College London)
Francisco Casacuberta (Universitat Politècnica de València)
Sheila Castilho (Dublin City University)
Daniel Cer (Google Research; University of California at Berkeley)
Boxing Chen (Alibaba)
Colin Cherry (Google)
Mara Chinea-Rios (Symanto Research)
Vishal Chowdhary (MSR)
Chenhui Chu (Kyoto University)
Josep Crego (SYSTRAN)
James Cross (Facebook)
Raj Dabre (NICT)
Steve DeNeefe (SDL Research)
Michael Denkowski (Amazon)
Mattia A. Di Gangi (AppTek GmbH)
Miguel Domingo (Universitat Politècnica de València)
Kevin Duh (Johns Hopkins University)
Hiroshi Echizen-ya (Hokkai-Gakuen University)
Sergey Edunov (Faceook AI Research)
Miquel Esplà-Gomis (Universitat d'Alacant)
Marcello Federico (Amazon AI)
Yang Feng (Institute of Computing Technology, Chinese Academy of Sciences)
Orhan Firat (Google AI)
Mikel L. Forcada (Universitat d'Alacant)
George Foster (Google)
Atsushi Fujita (National Institute of Information and Communications Technology)
Yang Gao (Institute of Software, Chinese Academy of Sciences)
Ulrich Germann (University of Edinburgh)
Jesús González-Rubio (WebInterpret)
Isao Goto (NHK)
Cyril Goutte (National Research Council Canada)
Roman Grundkiewicz (University of Edinburgh)
Mandy Guo (Google)
Jeremy Gwinnup (Air Force Research Laboratory)
Thanh-Le Ha (Karlsruhe Institute of Technology)
Greg Hanneman (Amazon)
Christian Hardmeier (Uppsala universitet/University of Edinburgh)
John Henderson (MITRE)
Christian Herold (RWTH Aachen University)
Felix Hieber (Amazon)
Almut Silja Hildebrand (Amazon)

Cong Duy Vu Hoang (Oracle)
Mika Hämäläinen (University of Helsinki, Rootroo Ltd)
Kenji Imamura (National Institute of Information and Communications Technology)
Aizhan Imankulova (Tokyo Metropolitan University)
Phillip Keung (Amazon)
Shahram Khadivi (eBay)
Huda Khayrallah (Johns Hopkins University)
Yunsu Kim (RWTH Aachen University)
Rebecca Knowles (National Research Council Canada)
Julia Kreutzer (Google)
Roland Kuhn (National Research Council of Canada)
Shankar Kumar (Google)
Anoop Kunchukuttan (Microsoft AI and Research)
Veronika Laippala (University of Turku)
Surafel Melaku Lakew (Amazon AI)
Ekaterina Lapshinova-Koltunski (Universität des Saarlandes)
Alon Lavie (Unbabel/Carnegie Mellon University)
Jing Li (Department of Computing, The Hong Kong Polytechnic University)
Jindřich Libovický (Ludwig Maximilian University of Munich)
Patrick Littell (National Research Council of Canada)
Fei Liu (University of Central Florida)
Qun Liu (Huawei Noah's Ark Lab)
Samuel Läubli (University of Zurich)
Vivien Macketanz (German Research Center for Artificial Intelligence (DFKI))
Gideon Maillette de Buy Wenniger (Bernoulli Institute for Mathematics, Computer Science and Artificial Intelligence, University of Groningen, Groningen, The Netherlands)
Andreas Maletti (Universität Leipzig)
Sameen Maruf (Monash University)
Arya D. McCarthy (Johns Hopkins University)
Antonio Valerio Miceli Barone (The University of Edinburgh)
Philippe Muller (IRIT, University of Toulouse)
Kenton Murray (Johns Hopkins University)
Tomáš Musil (Charles University)
Mathias Müller (University of Zurich)
Preslav Nakov (Qatar Computing Research Institute, HBKU)
Graham Neubig (Carnegie Mellon University)
Jan Niehues (Maastricht University)
Xing Niu (Amazon AI)
Tsuyoshi Okita (Kyushu institute of technology/RIKEN AIP)
Arturo Oncevay (The University of Edinburgh)
Carla Parra Escartín (Iconic Translation Machines)
Pavel Pecina (Charles University)
Stephan Peitz (Apple)
Sergio Penkale (Lingo24)
Mārcis Pinnis (Tilde)
Maja Popović (ADAPT Centre @ DCU)
Matīss Rikters (The University of Tokyo)

Annette Rios (University of Zurich)
Raphael Rubino (NICT)
Elizabeth Salesky (Johns Hopkins University)
Hassan Sawaf (aixplain, inc.)
Rico Sennrich (University of Zurich)
Aditya Siddhant (Google)
Patrick Simianer (Lilt)
Linfeng Song (Tencent AI Lab)
Felix Stahlberg (Google Research)
Dario Stojanovski (LMU Munich)
Katsuhito Sudoh (Nara Institute of Science and Technology (NAIST))
Víctor M. Sánchez-Cartagena (Universitat d'Alacant)
Aleš Tamchyna (Memsource)
Gongbo Tang (Uppsala University)
Brian Thompson (Johns Hopkins University)
Jörg Tiedemann (University of Helsinki)
Antonio Toral (University of Groningen)
Ke Tran (Amazon)
Ferhan Ture (Comcast Applied AI Research)
Masao Utiyama (NICT)
Dusan Varis (Charles University, Institute of Formal and Applied Linguistics)
David Vilar (Google)
Ekaterina Vylomova (University of Melbourne)
Weiyue Wang (RWTH Aachen University)
Taro Watanabe (Nara Institute of Science and Technology)
Hua Wu (Baidu)
Joern Wuebker (Lilt, Inc.)
Hainan Xu (Google)
Yinfei Yang (Google)
François Yvon (LIMSI/CNRS)
Xuan Zhang (Johns Hopkins University)
Zhong Zhou (Carnegie Mellon University)

Table of Contents

Conference Program

Thursday, November 19, 2020

9:45–10:00 ***Opening Remarks***

10:00–11:00 **Session 1: Shared Task Overview Papers I (Chair: Rachel Bawden)**

Findings of the 2020 Conference on Machine Translation (WMT20)
Loïc Barrault, Magdalena Biesialska, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Matthias Huck, Eric Joanis, Tom Kocmi, Philipp Koehn, Chi-kiu Lo, Nikola Ljubešić, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Santanu Pal, Matt Post and Marcos Zampieri

Findings of the First Shared Task on Lifelong Learning Machine Translation
Loïc Barrault, Magdalena Biesialska, Marta R. Costa-jussà, Fethi Bougares and Olivier Galibert

Findings of the WMT 2020 Shared Task on Chat Translation
M. Amin Farajian, António V. Lopes, André F. T. Martins, Sameen Maruf and Gholamreza Haffari

Findings of the WMT 2020 Shared Task on Machine Translation Robustness
Lucia Specia, Zhenhao Li, Juan Pino, Vishrav Chaudhary, Francisco Guzmán, Graham Neubig, Nadir Durrani, Yonatan Belinkov, Philipp Koehn, Hassan Sajjad, Paul Michel and Xian Li

11:00–12:30 **Session 2: Shared Task Posters I**

11:00–12:30 **News Translation Task**

11:00–12:30 *The University of Edinburgh's English-Tamil and English-Inuktitut Submissions to the WMT20 News Translation Task*
Rachel Bawden, Alexandra Birch, Radina Dobreva, Arturo Oncevay, Antonio Valerio Miceli Barone and Philip Williams

11:00–12:30 *GTCOM Neural Machine Translation Systems for WMT20*
Chao Bei, Hao Zong, Qingmin Liu and Conghu Yuan

11:00–12:30 *DiDi's Machine Translation System for WMT2020*
Tanfang Chen, Weiwei Wang, Wenyang Wei, Xing Shi, Xiangang Li, Jieping Ye and Kevin Knight

11:00–12:30 *Facebook AI's WMT20 News Translation Task Submission*
Peng-Jen Chen, Ann Lee, Changhan Wang, Naman Goyal, Angela Fan, Mary Williamson and Jiatao Gu

Thursday, November 19, 2020 (continued)

11:00–12:30 *Linguistically Motivated Subwords for English-Tamil Translation: University of Groningen's Submission to WMT-2020*
Prajit Dhar, Arianna Bisazza and Gertjan van Noord

11:00–12:30 *The TALP-UPC System Description for WMT20 News Translation Task: Multilingual Adaptation for Low Resource MT*
Carlos Escolano, Marta R. Costa-jussà and José A. R. Fonollosa

11:00–12:30 *An Iterative Knowledge Transfer NMT System for WMT20 News Translation Task*
Jiwan Kim, Soyoon Park, Sangha Kim and Yoonjung Choi

11:00–12:30 *Tohoku-AIP-NTT at WMT 2020 News Translation Task*
Shun Kiyono, Takumi Ito, Ryuto Konno, Makoto Morishita and Jun Suzuki

11:00–12:30 *NRC Systems for the 2020 Inuktitut-English News Translation Task*
Rebecca Knowles, Darlene Stewart, Samuel Larkin and Patrick Littell

11:00–12:30 *CUNI Submission for the Inuktitut Language in WMT News 2020*
Tom Kocmi

11:00–12:30 *Tilde at WMT 2020: News Task Systems*
Rihards Krišlauks and Mārcis Pinnis

11:00–12:30 *Samsung R&D Institute Poland submission to WMT20 News Translation Task*
Mateusz Krubiński, Marcin Chochowski, Bartłomiej Boczek, Mikołaj Koszowski, Adam Dobrowolski, Marcin Szymański and Paweł Przybysz

11:00–12:30 *Speed-optimized, Compact Student Models that Distill Knowledge from a Larger Teacher Model: the UEDIN-CUNI Submission to the WMT 2020 News Translation Task*
Ulrich Germann, Roman Grundkiewicz, Martin Popel, Radina Dobreva, Nikolay Bogoychev and Kenneth Heafield

11:00–12:30 *The University of Edinburgh's submission to the German-to-English and English-to-German Tracks in the WMT 2020 News Translation and Zero-shot Translation Robustness Tasks*
Ulrich Germann

11:00–12:30 *Contact Relatedness can help improve multilingual NMT: Microsoft STCI-MT @ WMT20*
Vikrant Goyal, Anoop Kunchukuttan, Rahul Kejriwal, Siddharth Jain and Amit Bhagwat

11:00–12:30 *The AFRL WMT20 News Translation Systems*
Jeremy Gwinnup and Tim Anderson

Thursday, November 19, 2020 (continued)

11:00–12:30 *The Ubiqus English-Inuktitut System for WMT20*
François Hernandez and Vincent Nguyen

11:00–12:30 *SJTU-NICT's Supervised and Unsupervised Neural Machine Translation Systems for the WMT20 News Translation Task*
Zuchao Li, Hai Zhao, Rui Wang, Kehai Chen, Masao Utiyama and Eiichiro Sumita

11:00–12:30 *Combination of Neural Machine Translation Systems at WMT20*
Benjamin Marie, Raphael Rubino and Atsushi Fujita

11:00–12:30 *WeChat Neural Machine Translation Systems for WMT20*
Fandong Meng, Jianhao Yan, Yijin Liu, Yuan Gao, Xianfeng Zeng, Qinsong Zeng, Peng Li, Ming Chen, Jie Zhou, Sifan Liu and Hao Zhou

11:00–12:30 *PROMT Systems for WMT 2020 Shared News Translation Task*
Alexander Molchanov

11:00–12:30 *eTranslation's Submissions to the WMT 2020 News Translation Task*
Csaba Oravecz, Katina Bontcheva, László Tihanyi, David Kolovratnik, Bhavani Bhaskar, Adrien Lardilleux, Szymon Klocek and Andreas Eisele

11:00–12:30 *The ADAPT System Description for the WMT20 News Translation Task*
Venkatesh Parthasarathy, Akshai Ramesh, Rejwanul Haque and Andy Way

11:00–12:30 *CUNI English-Czech and English-Polish Systems in WMT20: Robust Document-Level Training*
Martin Popel

11:00–12:30 *Machine Translation for English–Inuktitut with Segmentation, Data Acquisition and Pre-Training*
Christian Roest, Lukas Edman, Gosse Minnema, Kevin Kelly, Jennifer Spenader and Antonio Toral

11:00–12:30 *OPPO's Machine Translation Systems for WMT20*
Tingxun Shi, Shiyu Zhao, Xiaopu Li, Xiaoxue Wang, Qian Zhang, Di Ai, Dawei Dang, Xue Zhengshan and JIE HAO

11:00–12:30 *HW-TSC's Participation in the WMT 2020 News Translation Shared Task*
Daimeng Wei, Hengchao Shang, Zhanglin Wu, Zhengzhe Yu, Liangyou Li, Jiaxin Guo, Minghan Wang, Hao Yang, Lizhi Lei, Ying Qin and Shiliang Sun

11:00–12:30 *IIE's Neural Machine Translation Systems for WMT20*
Xiangpeng Wei, Ping Guo, Yunpeng Li, Xingsheng Zhang, Luxi Xing and Yue Hu

Thursday, November 19, 2020 (continued)

11:00–12:30 *The Volctrans Machine Translation System for WMT20*
Liwei Wu, Xiao Pan, Zehui Lin, Yaoming ZHU, Mingxuan Wang and Lei Li

11:00–12:30 *Tencent Neural Machine Translation Systems for the WMT20 News Translation Task*
Shuangzhi Wu, Xing Wang, Longyue Wang, Fangxu Liu, Jun Xie, Zhaopeng Tu, Shuming Shi and Mu Li

11:00–12:30 *Russian-English Bidirectional Machine Translation System*
ariel Xv

11:00–12:30 *The DeepMind Chinese–English Document Translation System at WMT2020*
Lei Yu, Laurent Sartran, Po-Sen Huang, Wojciech Stokowiec, Domenic Donato, Srivatsan Srinivasan, Alek Andreev, Wang Ling, Sona Mokra, Agustin Dal Lago, Yotam Doron, Susannah Young, Phil Blunsom and Chris Dyer

11:00–12:30 *The NiuTrans Machine Translation Systems for WMT20*
Yuhao Zhang, Ziyang Wang, Runzhe Cao, Binghao Wei, Weiqiao Shan, Shuhan Zhou, Abudurexiti Reheman, Tao Zhou, Xin Zeng, Laohu Wang, Yongyu Mu, Jingnan Zhang, Xiaoqian Liu, Xuanjun Zhou, Yinqiao Li, Bei Li, Tong Xiao and Jingbo Zhu

11:00–12:30 Test Sets

11:00-12:30 *Fine-grained linguistic evaluation for state-of-the-art Machine Translation*
Eleftherios Avramidis, Vivien Macketanz, Ursula Strohriegel, Aljoscha Burchardt and Sebastian Möller

11:00–12:30 *Gender Coreference and Bias Evaluation at WMT 2020*
Tom Kocmi, Tomasz Limisiewicz and Gabriel Stanovsky

11:00–12:30 *The MUCOW word sense disambiguation test suite at WMT 2020*
Yves Scherrer, Alessandro Raganato and Jörg Tiedemann

11:00–12:30 *WMT20 Document-Level Markable Error Exploration*
Vilém Zouhar, Tereza Vojtěchová and Ondřej Bojar

Thursday, November 19, 2020 (continued)

11:00–12:30 Similar Language Translation Task

11:00–12:30 *Translating Similar Languages: Role of Mutual Intelligibility in Multilingual Transformers*
Ife Adebara, El Moatez Billah Nagoudi and Muhammad Abdul Mageed

11:00–12:30 *Attention Transformer Model for Translation of Similar Languages*
Farhan Dhanani and Muhammad Rafi

11:00–12:30 *Transformer-based Neural Machine Translation System for Hindi – Marathi: WMT20 Shared Task*
Amit Kumar, Rupjyoti Baruah, Rajesh Kumar Mundotiya and Anil Kumar Singh

11:00–12:30 *Hindi-Marathi Cross Lingual Model*
Sahinur Rahman Laskar, Abdullah Faiz Ur Rahman Khilji, Partha Pakray and Sivaji Bandyopadhyay

11:00–12:30 *Transfer Learning for Related Languages: Submissions to the WMT20 Similar Language Translation Task*
Lovish Madaan, Soumya Sharma and Parag Singla

11:00–12:30 *The IPN-CIC team system submission for the WMT 2020 similar language task*
Luis A. Menéndez-Salazar, Grigori Sidorov and Marta R. Costa-Jussà

11:00–12:30 *NMT based Similar Language Translation for Hindi - Marathi*
Vandan Mujadia and Dipti Sharma

11:00–12:30 *NUIG-Panlingua-KMI Hindi-Marathi MT Systems for Similar Language Translation Task @ WMT 2020*
Atul Kr. Ojha, Priya Rani, Akanksha Bansal, Bharathi Raja Chakravarthi, Ritesh Kumar and John P. McCrae

11:00–12:30 *Neural Machine Translation for Similar Languages: The Case of Indo-Aryan Languages*
Santanu Pal and Marcos Zampieri

11:00–12:30 *Neural Machine Translation between similar South-Slavic languages*
Maja Popović and Alberto Poncelas

11:00–12:30 *Infosys Machine Translation System for WMT20 Similar Language Translation Task*
Kamalkumar Rathinasamy, Amanpreet Singh, Balaguru Sivasambagupta, Prajna Prasad Neerchal and Vani Sivasankaran

Thursday, November 19, 2020 (continued)

11:00–12:30 *Document Level NMT of Low-Resource Languages with Backtranslation*
Sami Ul Haq, Sadaf Abdul Rauf, Arsalan Shaukat and Abdullah Saeed

11:00–12:30 *Multilingual Neural Machine Translation: Case-study for Catalan, Spanish and Portuguese Romance Languages*
Pere Vergés Boncompte and Marta R. Costa-jussà

11:00–12:30 *A3-108 Machine Translation System for Similar Language Translation Shared Task 2020*
Saumitra Yadav and Manish Shrivastava

11:00–12:30 Chat Translation Task

11:00–12:30 *The University of Maryland's Submissions to the WMT20 Chat Translation Task: Searching for More Data to Adapt Discourse-Aware Neural Machine Translation*
Calvin Bao, Yow-Ting Shiue, Chujun Song, Jie Li and Marine Carpuat

11:00–12:30 *Naver Labs Europe's Participation in the Robustness, Chat, and Biomedical Tasks at WMT 2020*
Alexandre Berard, Ioan Calapodescu, Vassilina Nikoulina and Jerin Philip

11:00–12:30 *The University of Edinburgh-Uppsala University's Submission to the WMT 2020 Chat Translation Task*
Nikita Moghe, Christian Hardmeier and Rachel Bawden

11:00–12:30 *JUST System for WMT20 Chat Translation Task*
Roweida Mohammed, Mahmoud Al-Ayyoub and Malak Abdullah

11:00–12:30 *Tencent AI Lab Machine Translation Systems for WMT20 Chat Translation Task*
Longyue Wang, Zhaopeng Tu, Xing Wang, Li Ding, Liang Ding and Shuming Shi

12:30–13:00 ***Break***

Thursday, November 19, 2020 (continued)

13:00–14:00 Session 3: Research Papers I (Chair: Tom Kocmi)

Combining Sequence Distillation and Transfer Learning for Efficient Low-Resource Neural Machine Translation Models
Raj Dabre and Atsushi Fujita

Fast Interleaved Bidirectional Sequence Generation
Biao Zhang, Ivan Titov and Rico Sennrich

Priming Neural Machine Translation
Minh Quang Pham, Jitao Xu, Josep Crego, François Yvon and Jean Senellart

Subword Segmentation and a Single Bridge Language Affect Zero-Shot Neural Machine Translation
Annette Rios, Mathias Müller and Rico Sennrich

Look It Up: Bilingual and Monolingual Dictionaries Improve Neural Machine Translation
Xing Jie Zhong and David Chiang

14:00–16:00 ***Break***

16:00–17:00 Session 4: Shared Task Overview I (Chair: Antonio Toral)

17:00–18:30 Session 5: Shared Task Posters I

18:30–19:00 ***Break***

Thursday, November 19, 2020 (continued)

19:00–20:00 Session 6: Research Papers II (Chair: Colin Cherry)

Complete Multilingual Neural Machine Translation
Markus Freitag and Orhan Firat

Paraphrase Generation as Zero-Shot Multilingual Translation: Disentangling Semantic Similarity from Lexical and Syntactic Diversity
Brian Thompson and Matt Post

When Does Unsupervised Machine Translation Work?
Kelly Marchisio, Kevin Duh and Philipp Koehn

Language Models not just for Pre-training: Fast Online Neural Noisy Channel Modeling
Shruti Bhosale, Kyra Yee, Sergey Edunov and Michael Auli

Friday, November 20, 2020

9:00–10:00 Session 7: Research Papers III (Chair: Marta R. Costa-jussà)

Towards Multimodal Simultaneous Neural Machine Translation
Aizhan Imankulova, Masahiro Kaneko, Tosho Hirasawa and Mamoru Komachi

Diving Deep into Context-Aware Neural Machine Translation
Jingjing Huo, Christian Herold, Yingbo Gao, Leonard Dahlmann, Shahram Khadivi and Hermann Ney

A Study of Residual Adapters for Multi-Domain Neural Machine Translation
Minh Quang Pham, Josep Maria Crego, François Yvon and Jean Senellart

Mitigating Gender Bias in Machine Translation with Target Gender Annotations
Artūrs Stafanovičs, Mārcis Pinnis and Toms Bergmanis

Document-aligned Japanese-English Conversation Parallel Corpus
Matīss Rikters, Ryokan Ri, Tong Li and Toshiaki Nakazawa

Friday, November 20, 2020 (continued)

10:00–11:00 Session 8: Shared Task Overview Papers II (Chair Jindřich Libovický)

Findings of the WMT 2020 Shared Task on Automatic Post-Editing
Rajen Chatterjee, Markus Freitag, Matteo Negri and Marco Turchi

Findings of the WMT 2020 Biomedical Translation Shared Task: Basque, Italian and Russian as New Additional Languages
Rachel Bawden, Giorgio Maria Di Nunzio, Cristian Grozea, Inigo Jauregi Unanue, Antonio Jimeno Yepes, Nancy Mah, David Martinez, Aurélie Névéol, Mariana Neves, Maite Oronoz, Olatz Perez-de-Viñaspre, Massimo Piccardi, Roland Roller, Amy Siu, Philippe Thomas, Federica Vezzani, Maika Vicente Navarro, Dina Wiemann and Lana Yeganova

Results of the WMT20 Metrics Shared Task
Nitika Mathur, Johnny Wei, Markus Freitag, Qingsong Ma and Ondřej Bojar

Findings of the WMT 2020 Shared Task on Parallel Corpus Filtering and Alignment
Philipp Koehn, Vishrav Chaudhary, Ahmed El-Kishky, Naman Goyal, Peng-Jen Chen and Francisco Guzmán

Findings of the WMT 2020 Shared Task on Quality Estimation
Lucia Specia, Frédéric Blain, Marina Fomicheva, Erick Fonseca, Vishrav Chaudhary, Francisco Guzmán and André F. T. Martins

Findings of the WMT 2020 Shared Tasks in Unsupervised MT and Very Low Resource Supervised MT
Alexander Fraser

11:00–12:30 Session 9: Shared Task Posters II

Automatic Post-Editing Task

11:00–12:30 *Cross-Lingual Transformers for Neural Automatic Post-Editing*
Dongjun Lee

11:00–12:30 *POSTECH-ETRI's Submission to the WMT2020 APE Shared Task: Automatic Post-Editing with Cross-lingual Language Model*
Jihyung Lee, WonKee Lee, Jaehun Shin, Baikjin Jung, Young-Kil Kim and Jong-Hyeok Lee

11:00–12:30 *Noising Scheme for Data Augmentation in Automatic Post-Editing*
WonKee Lee, Jaehun Shin, Baikjin Jung, Jihyung Lee and Jong-Hyeok Lee

11:00–12:30 *Alibaba's Submission for the WMT 2020 APE Shared Task: Improving Automatic Post-Editing with Pre-trained Conditional Cross-Lingual BERT*
Jiayi Wang, Ke Wang, Kai Fan, Yuqi Zhang, Jun Lu, Xin Ge, Yangbin Shi and Yu Zhao

11:00–12:30 *HW-TSC's Participation at WMT 2020 Automatic Post Editing Shared Task*
Hao Yang, Minghan Wang, Daimeng Wei, Hengchao Shang, Jiaxin Guo, Zongyao Li, Lizhi Lei, Ying Qin, Shimin Tao, Shiliang Sun and Yimeng Chen

Biomedical Translation Task

11:00–12:30 *LIMSI @ WMT 2020*
Sadaf Abdul Rauf, José Carlos Rosales Núñez, Minh Quang Pham and François Yvon

11:00–12:30 *Elhuyar submission to the Biomedical Translation Task 2020 on terminology and abstracts translation*
Ander Corral and Xabier Saralegi

11:00–12:30 *YerevaNN's Systems for WMT20 Biomedical Translation Task: The Effect of Fixing Misaligned Sentence Pairs*
Karen Hambardzumyan, Hovhannes Tamoyan and Hrant Khachatrian

11:00–12:30 *Pretrained Language Models and Backtranslation for English-Basque Biomedical Neural Machine Translation*
Inigo Jauregi Unanue and Massimo Piccardi

11:00–12:30 *Lite Training Strategies for Portuguese-English and English-Portuguese Translation*
Alexandre Lopes, Rodrigo Nogueira, Roberto Lotufo and Helio Pedrini

Friday, November 20, 2020 (continued)

11:00–12:30 *The ADAPT's Submissions to the WMT20 Biomedical Translation Task*
Prashant Nayak, Rejwanul Haque and Andy Way

11:00–12:30 *FJWU participation for the WMT20 Biomedical Translation Task*
Sumbal Naz, Sadaf Abdul Rauf, Noor-e- Hira and Sami Ul Haq

11:00–12:30 *Huawei's Submissions to the WMT20 Biomedical Translation Task*
Wei Peng, Jianfeng Liu, Minghan Wang, Liangyou Li, Xupeng Meng, Hao Yang and Qun Liu

11:00–12:30 *Addressing Exposure Bias With Document Minimum Risk Training: Cambridge at the WMT20 Biomedical Translation Task*
Danielle Saunders and Bill Byrne

11:00–12:30 *UoS Participation in the WMT20 Translation of Biomedical Abstracts*
Felipe Soares and Delton Vaz

11:00–12:30 *Ixamed's submission description for WMT20 Biomedical shared task: benefits and limitations of using terminologies for domain adaptation*
Xabier Soto, Olatz Perez-de-Viñaspre, Gorka Labaka and Maite Oronoz

11:00–12:30 *Tencent AI Lab Machine Translation Systems for the WMT20 Biomedical Translation Task*
Xing Wang, Zhaopeng Tu, Longyue Wang and Shuming Shi

Metrics Task

11:00–12:30 *ParBLEU: Augmenting Metrics with Automatic Paraphrases for the WMT'20 Metrics Shared Task*
Rachel Bawden, Biao Zhang, Andre Tättar and Matt Post

11:00–12:30 *Extended Study on Using Pretrained Language Models and YiSi-1 for Machine Translation Evaluation*
Chi-kiu Lo

11:00–12:30 *Machine Translation Reference-less Evaluation using YiSi-2 with Bilingual Mappings of Massive Multilingual Language Model*
Chi-kiu Lo and Samuel Larkin

11:00–12:30 *Unbabel's Participation in the WMT20 Metrics Shared Task*
Ricardo Rei, Craig Stewart, Ana C Farinha and Alon Lavie

Friday, November 20, 2020 (continued)

11:00–12:30 *Learning to Evaluate Translation Beyond English: BLEURT Submissions to the WMT Metrics 2020 Shared Task*
Thibault Sellam, Amy Pu, Hyung Won Chung, Sebastian Gehrmann, Qijun Tan, Markus Freitag, Dipanjan Das and Ankur Parikh

11:00–12:30 *Towards a Better Evaluation of Metrics for Machine Translation*
Peter Stanchev, Weiyue Wang and Hermann Ney

11:00–12:30 *Incorporate Semantic Structures into Machine Translation Evaluation via UCCA*
Jin Xu, Yinuo Guo and Junfeng Hu

Parallel Corpus Filtering Task

11:00–12:30 *Filtering Noisy Parallel Corpus using Transformers with Proxy Task Learning*
Haluk Açarçiçek, Talha Çolakoğlu, pınar ece aktan hatipoğlu, Chong Hsuan Huang and Wei Peng

11:00–12:30 *Score Combination for Improved Parallel Corpus Filtering for Low Resource Conditions*
Muhammad ElNokrashy, Amr Hendy, Mohamed Abdelghaffar, Mohamed Afify, Ahmed Tawfik and Hany Hassan Awadalla

11:00–12:30 *Bicleaner at WMT 2020: Universitat d'Alacant-Prompsit's submission to the parallel corpus filtering shared task*
Miquel Esplà-Gomis, Víctor M. Sánchez-Cartagena, Jaume Zaragoza-Bernabeu and Felipe Sánchez-Martínez

11:00–12:30 *An exploratory approach to the Parallel Corpus Filtering shared task WMT20*
Ankur Kejriwal and Philipp Koehn

11:00–12:30 *Dual Conditional Cross Entropy Scores and LASER Similarity Scores for the WMT20 Parallel Corpus Filtering Shared Task*
Felicia Koerner and Philipp Koehn

11:00–12:30 *Improving Parallel Data Identification using Iteratively Refined Sentence Alignments and Bilingual Mappings of Pre-trained Language Models*
Chi-kiu Lo and Eric Joanis

11:00–12:30 *Alibaba Submission to the WMT20 Parallel Corpus Filtering Task*
Jun Lu, Xin Ge, Yangbin Shi and Yuqi Zhang

11:00–12:30 *Volctrans Parallel Corpus Filtering System for WMT 2020*
Runxin Xu, Zhuo Zhi, Jun Cao, Mingxuan Wang and Lei Li

Quality Estimation Task

11:00–12:30 *PATQUEST: Papago Translation Quality Estimation*
Yujin Baek, Zae Myung Kim, Jihyung Moon, Hyunjoong Kim and Eunjeong Park

11:00–12:30 *RTM Ensemble Learning Results at Quality Estimation Task*
Ergun Biçici

11:00–12:30 *NJU's submission to the WMT20 QE Shared Task*
Qu Cui, Xiang Geng, Shujian Huang and Jiajun CHEN

11:00–12:30 *BERGAMOT-LATTE Submissions for the WMT20 Quality Estimation Shared Task*
Marina Fomicheva, Shuo Sun, Lisa Yankovskaya, Frédéric Blain, Vishrav Chaudhary, Mark Fishel, Francisco Guzmán and Lucia Specia

11:00–12:30 *The NiuTrans System for the WMT20 Quality Estimation Shared Task*
Chi Hu, Hui Liu, Kai Feng, Chen Xu, Nuo Xu, Zefan Zhou, Shiqin Yan, Yingfeng Luo, Chenglong Wang, Xia Meng, Tong Xiao and Jingbo Zhu

11:00–12:30 *Two-Phase Cross-Lingual Language Model Fine-Tuning for Machine Translation Quality Estimation*
Dongjun Lee

11:00–12:30 *IST-Unbabel Participation in the WMT20 Quality Estimation Shared Task*
João Moura, miguel vera, Daan van Stigt, Fabio Kepler and André F. T. Martins

11:00–12:30 *TMUOU Submission for WMT20 Quality Estimation Shared Task*
Akifumi Nakamachi, Hiroki Shimanaka, Tomoyuki Kajiwara and Mamoru Komachi

11:00–12:30 *NICT Kyoto Submission for the WMT'20 Quality Estimation Task: Intermediate Training for Domain and Task Adaptation*
Raphael Rubino

11:00–12:30 *TransQuest at WMT2020: Sentence-Level Direct Assessment*
Tharindu Ranasinghe, Constantin Orasan and Ruslan Mitkov

11:00–12:30 *HW-TSC's Participation at WMT 2020 Quality Estimation Shared Task*
Minghan Wang, Hao Yang, Hengchao Shang, Daimeng Wei, Jiaxin Guo, Lizhi Lei, Ying Qin, Shimin Tao, Shiliang Sun, Yimeng Chen and Liangyou Li

Friday, November 20, 2020 (continued)

11:00–12:30 *Tencent submission for WMT20 Quality Estimation Shared Task*
Haijiang Wu, Zixuan Wang, Qingsong Ma, Xinjie Wen, Ruichen Wang, Xiaoli Wang, Yulin Zhang, Zhipeng Yao and Siyao Peng

11:00–12:30 *Zero-Shot Translation Quality Estimation with Explicit Cross-Lingual Patterns*
Lei Zhou, Liang Ding and Koichi Takeda

Unsupervised and Very Low-Resource Translation Task

11:00–12:30 *NLPRL System for Very Low Resource Supervised Machine Translation*
Rupjyoti Baruah, Rajesh Kumar Mundotiya, Amit Kumar and Anil kumar Singh

11:00–12:30 *Low-Resource Translation as Language Modeling*
Tucker Berckmann and Berkan Hiziroglu

11:00–12:30 *The LMU Munich System for the WMT 2020 Unsupervised Machine Translation Shared Task*
Alexandra Chronopoulou, Dario Stojanovski, Viktor Hangya and Alexander Fraser

11:00–12:30 *UdS-DFKI@WMT20: Unsupervised MT and Very Low Resource Supervised MT for German-Upper Sorbian*
Sourav Dutta, Jesujoba Alabi, Saptarashmi Bandyopadhyay, Dana Ruiter and Josef van Genabith

11:00–12:30 *Data Selection for Unsupervised Translation of German–Upper Sorbian*
Lukas Edman, Antonio Toral and Gertjan van Noord

11:00–12:30 *The LMU Munich System for the WMT20 Very Low Resource Supervised MT Task*
Jindřich Libovický, Viktor Hangya, Helmut Schmid and Alexander Fraser

11:00–12:30 *NRC Systems for Low Resource German-Upper Sorbian Machine Translation 2020: Transfer Learning with Lexical Modifications*
Rebecca Knowles, Samuel Larkin, Darlene Stewart and Patrick Littell

11:00–12:30 *CUNI Systems for the Unsupervised and Very Low Resource Translation Task in WMT20*
Ivana Kvapilíková, Tom Kocmi and Ondřej Bojar

11:00–12:30 *The University of Helsinki and Aalto University submissions to the WMT 2020 news and low-resource translation tasks*
Yves Scherrer, Stig-Arne Grönroos and Sami Virpioja

Friday, November 20, 2020 (continued)

11:00–12:30 *The NITS-CNLP System for the Unsupervised MT Task at WMT 2020*
Salam Michael Singh, Thoudam Doren Singh and Sivaji Bandyopadhyay

11:00–12:30 *Adobe AMPS's Submission for Very Low Resource Supervised Translation Task at WMT20*
Keshaw Singh

12:30–13:00 ***Break***

13:00–14:00 **Session 10: Invited Talk: "Low-resourcedness" Beyond Data**

Ignatius Ezeani, Jade Abbott, Julia Kreutzer, Salomon Kabongo, Perez Ogayo, Shamsuddeen Hassan Muhammad, Rubungo Andre Niyongabo, Jamiil Toure Ali, Kathleen Siminyu, Salomey Osei, Wilhelmina Nekoto, Arshath Ramkilowan, Masabata Mokgesi-Selinga, Bonaventure Dossou, Ayodele Olabiyi, Blessing Sibanda, Akinola Oluwole, Vukosi Marivate, Orevaoghene Ahia

14:00–15:30 **Session 11: Panel Discussion (Moderator: Lexi Birch)**

Panel: Jade Abbott, Anoop Kunchukuttan, Kathleen Siminyu and Jörg Tiedemann

15:30–16:00 ***Break***

16:00–17:00 **Session 12: Shared Task Overview II (Chair: Matt Post)**

Friday, November 20, 2020 (continued)

17:00–18:30 Session 13: Shared Task Posters II

18:30–19:00 ***Break***

19:00–20:00 Session 14: Research Papers IV (Chair: Michael Auli)

On the Same Page? Comparing Inter-Annotator Agreement in Sentence and Document Level Human Machine Translation Evaluation
Sheila Castilho

How Should Markup Tags Be Translated?
Greg Hanneman and Georgiana Dinu

The Tatoeba Translation Challenge – Realistic Data Sets for Low Resource and Multilingual MT
Jörg Tiedemann

Human-Paraphrased References Improve Neural Machine Translation
Markus Freitag, George Foster, David Grangier and Colin Cherry

Incorporating Terminology Constraints in Automatic Post-Editing
David Wan, Chris Kedzie, Faisal Ladhak, Marine Carpuat and Kathleen McKeown

Findings of the 2020 Conference on Machine Translation (WMT20)

Loïc Barrault University of Sheffield
Magdalena Biesialska UPC
Ondřej Bojar Charles University
Marta R. Costa-jussà UPC
Christian Federmann Microsoft Cloud + AI
Yvette Graham Trinity College Dublin
Roman Grundkiewicz Microsoft
Barry Haddow University of Edinburgh
Matthias Huck SAP SE
Eric Joanis NRC
Tom Kocmi Microsoft
Philipp Koehn JHU
Chi-kiu Lo NRC
Nikola Ljubešić Josef Stefan Institute
Christof Monz University of Amsterdam
Makoto Morishita NTT
Masaaki Nagata NTT
Toshiaki Nakazawa University of Tokyo
Santanu Pal WIPRO AI
Matt Post JHU
Marcos Zampieri Rochester Institute of Technology

Abstract

This paper presents the results of the news translation task and the similar language translation task, both organised alongside the Conference on Machine Translation (WMT) 2020. In the news task, participants were asked to build machine translation systems for any of 11 language pairs, to be evaluated on test sets consisting mainly of news stories. The task was also opened up to additional test suites to probe specific aspects of translation. In the similar language translation task, participants built machine translation systems for translating between closely related pairs of languages.

1 Introduction

The Fifth Conference on Machine Translation (WMT20)[1] was held online with EMNLP 2020 and hosted a number of shared tasks on various aspects of machine translation. This conference built on 14 previous editions of WMT as workshops and conferences (Koehn and Monz, 2006; Callison-Burch et al., 2007, 2008, 2009, 2010, 2011, 2012; Bojar et al., 2013, 2014, 2015, 2016, 2017, 2018; Barrault et al., 2019).

This year we conducted several official tasks. We report in this paper on the news and similar translation tasks. Additional shared tasks are described in separate papers in these proceedings:

- automatic post-editing (Chatterjee et al., 2020)
- biomedical translation (Bawden et al., 2020b)
- chat translation (Farajian et al., 2020)
- lifelong learning (Barrault et al., 2020)
- metrics (Mathur et al., 2020)
- parallel corpus filtering (Koehn et al., 2020)
- quality estimation (Specia et al., 2020a)
- robustness (Specia et al., 2020b)
- unsupervised and very low-resource translation (Fraser, 2020)

In the news translation task (Section 2), participants were asked to translate a shared test set, optionally restricting themselves to the provided training data ("constrained" condition). We included 22 translation directions this year, with translation between English and each of Chinese, Czech, German and Russian, as well as French to and from German being repeated from last year, and English to and from Inuktitut, Japanese, Polish and Tamil being new for this year. Furthermore, English to and from Khmer and Pashto were included, using the same test sets as in the corpus filtering task. The translation tasks covered a range of language families, and included both low-resource and high-resource pairs. System outputs for each task were evaluated both automatically and manually, but we only include the manual evaluation here.

The human evaluation (Section 3) involves asking human judges to score sentences output by anonymized systems. We obtained large numbers of assessments from researchers who contributed evaluations proportional to the number of tasks they entered. In addition, we used Mechanical Turk to collect further evaluations, as well as a pool of linguists. This year, the official manual evaluation metric is again based on judgments of adequacy on a 100-point scale, a method (known

[1] http://www.statmt.org/wmt20/

Proceedings of the 5th Conference on Machine Translation (WMT), pages 1–55

as "direct assessment") that we explored in the previous years with convincing results in terms of the trade-off between annotation effort and reliable distinctions between systems.

The primary objectives of WMT are to evaluate the state of the art in machine translation, to disseminate common test sets and public training data with published performance numbers, and to refine evaluation and estimation methodologies for machine translation. As before, all of the data, translations, and collected human judgments are publicly available.[2] We hope these datasets serve as a valuable resource for research into data-driven machine translation, automatic evaluation, or prediction of translation quality. News translations are also available for interactive visualization and comparison of differences between systems at http://wmt.ufal.cz/ using MT-ComparEval (Sudarikov et al., 2016).

In order to gain further insight into the performance of individual MT systems, we organized a call for dedicated "test suites", each focusing on some particular aspect of translation quality. A brief overview of the test suites is provided in Section 4.

Following the success of the first Similar Language Translation (SLT) task at WMT 2019 and the interest of the community in this topic (Costa-jussà et al., 2018; Popović et al., 2020), we organize a second iteration of the SLT task at WMT 2020. The goal of the shared task is to evaluate the performance of state-of-the-art MT systems on translating between pairs of closely-related languages from the same language family. SLT 2020 features five pairs of similar languages from three language families: Indo-Aryan (Hindi and Marathi), Romance (Catalan, Spanish, and Portuguese), and South-Slavic (Croatian, Serbian, and Slovene). Translations were evaluated in both directions using three automatic metrics: BLEU, RIBES, and TER. Results and main findings of the SLT shared task are discussed in Section 5.

2 News Translation Task

This recurring WMT task assesses the quality of MT on news domain text. As in the previous year, we included Chinese, Czech, German and Russian (into and out of English) as well as French-German. New language pairs for this year were Inuktitut, Japanese, Polish and Tamil (to and from English). We also included the two language pairs from the corpus filtering task (Pashto→English and Khmer→English), to give participants the opportunity to build and test MT systems using the large noisy corpora released for that task.

2.1 Test Data

As in previous years, the test sets consist (as far as possible) of unseen translations prepared specially for the task. The test sets are publicly released to be used as translation benchmarks in the coming years. Here we describe the production and composition of the test sets.

The test sets differed along several dimensions, which we list in Table 1. The differing aspects of the test sets are as follows:

Domain Most test sets are drawn from the "news" domain, which means the source texts were extracted from online news websites, and the translations were produced specifically for the task. The Pashto→English and Khmer→English test sets were drawn from wikipedia and, as last year, the French↔German test sets concentrated on EU-related news.

Due to limited resources and data available, the Inuktitut↔English test sets contain document- and sentence-aligned data collected from two domains: news and parliamentary. The news data were extracted from the Nunatsiaq News online news website. The parliamentary data were debates from the Nunavut Hansard that are more recent than the training corpus.

Development? For new languages we released a development set, produced in the same way as the test set.

Sentence-split? For some pairs, we did not sentence-split the source texts. In these cases, we extracted the text from the HTML source with paragraph breaks retained, and asked translators to maintain only the paragraph breaks. This was done in order to try to improve the quality of the human translation by allowing the translators more freedom. Some analysis of the paragraph-split pairs is presented in Section 2.1.1.

Directional? For most language pairs the source-side of the test set is the original, and the target-side of the test set is the translation. This is in contrast to the situation up until 2018 when our test sets were constructed from both "source-original" and "target-original" parts. Where a

[2] http://statmt.org/wmt20/results.html

development set is provided, it is a mixture of both "source-original" and "target-original" texts, in order to maximise its size, although the original language is always marked in the sgm file, except for Inuktitut↔English. The consequences of directionality in test sets has been discussed recently in the literature (Freitag et al., 2019; Laubli et al., 2020; Graham et al., 2020), and the conclusion is that it can have an effect on detrimental effect on the accuracy of system evaluation. We use "source-original" parallel sentences wherever possible, on the basis that it is the more realistic scenario for practical MT usage.

Exception: the test sets for the two Inuktitut↔English translation directions contain the same data, without regard to original direction. For most news text in the test and development sets, English was the original language and Inuktitut the translation, while the parliamentary data mixes the two directions.

The origins of the news test documents is shown in Table 5, and the size of the test sets in terms of sentence pairs and words is given in Figure 4. We generally aimed for 1000 sentences for a new language pair, and 2000 sentences for a previously used language pair (since there was no need to create a development set for a previously-used language pair). For test sets where the source was not sentence-split (see below) we aimed for an equivalent to 2000 sentences, but in running words.

In order to improve the consistency and quality of the test set translations, this year we prepared common translator briefs to be sent to each agency we used. We show the translator briefs in Appendix B (for sentence-split sources) and Appendix C (for paragraph-split sources).

2.1.1 Paragraph-split Test Sets

For the language pairs English↔Czech, English↔German and English→Chinese, we provided the translators with paragraph-split texts, instead of sentence-split texts. We did this in order to provide the translators with greater freedom and, hopefully, to improve the quality of the translation. Allowing translators to merge and split sentences removes one of the "translation shifts" identified by Popovic (2019), which can make translations create solely for MT evaluation different from translations produced for other purposes.

We first show some descriptive statistics of the source texts, for Czech, English and German, in Table 2, where we used the Moses sentence splitter (Koehn et al., 2007) to provide sentence boundaries. We can see that the number of sentences per paragraph is much lower for English, where in fact 70% of paragraphs only have single sentence. For Czech and German, the mean sentences per paragraph is quite similar (2.62 vs. 2.52).

The main question though, is whether translators tended to preserve the sentence structure when translating. To determine this, we split both source paragraphs and translations into sentence, and aligned them using hunalign (Varga et al., 2005) with the bitextor dictionaries (Esplà-Gomis, 2009). In Table 4 we show the counts of 1-1 sentence alignments, as well as cases where the translator merged or split neighbouring sentences. Note that these counts are approximate, since they are affected by errors in the automatic splitting and alignment.

Looking through examples of merges and splits, we see that most of them are relatively simple changes, where the translator has merged to clauses into a sentence, or split a sentence to clauses. Examples of such merges and splits are shown in Table 3, where the first and second are simple merges or splits, whereas the third is a rare case of more complex reordering. We leave a detailed analysis of the translators' treatment of paragraph-split data for future work.

2.2 Training Data

As in past years we provided a selection of parallel and monolingual corpora for model training, and development sets to tune system parameters. Participants were permitted to use any of the provided corpora to train systems for any of the language pairs. As well as providing updates on many of the previously released data sets, we included several new data sets, mainly to support the new language pairs. These included Wikimatrix (Schwenk et al., 2019), which was added for all language pairs where it was available. The news commentary and europarl corpora that we have been using since the earliest news task now have "data sheets", describing the data sets in standardised format (Costa-jussà et al., 2020).

For Tamil-English, we additionally included some recently crawled multilingual parallel corpora from Indian government websites (Haddow and Kirefu, 2020; Siripragada et al., 2020), the Tanzil corpus (Tiedemann, 2009), the Pavlick dic-

Europarl Parallel Corpus

	Czech ↔ English		German ↔ English		Polish↔ English		German ↔ French	
Sentences	645,241		1,825,745		632,435		1,801,076	
Words	14,948,900	17,380,340	48,125,573	50,506,059	14,691,199	16,995,232	47,517,102	55,366,136
Distinct words	172,452	63,289	371,748	113,960	170,271	62,694	368,585	134,762

News Commentary Parallel Corpus

	Czech ↔ English		German ↔ English		Russian ↔ English	
Sentences	248,927		361,735		308,853	
Words	5,570,734	6,156,063	9,199,170	9,127,331	7,867,940	8,200,081
Distinct words	174,952	70,115	206,506	83,701	201,616	80,219

	Chinese ↔ English		Japanese ↔ English		German ↔ French	
Sentences	312,489		1,818		276,637	
Words	–	7,939,817	–	44,418	7,148,178	8,703,088
Distinct words	–	76,013	–	6,165	178,453	85,189

Common Crawl Parallel Corpus

	German ↔ English		Czech ↔ English		Russian ↔ English		French ↔ German	
Sentences	2,399,123		161,838		878,386		622,288	
Words	54,575,405	58,870,638	3,529,783	3,927,378	21,018,793	21,535,122	13,991,973	12,217,457
Distinct words	1,640,835	823,480	210,170	128,212	764,203	432,062	676,725	932,137

ParaCrawl Parallel Corpus

	German ↔ English		Czech ↔ English		Polish ↔ English	
Sentences	34,371,306		5,345,693		6,577,804	
Words	767,321,987	813,326,217	115,294,152	124,695,776	151,873,495	167,023,296
Distinct Words	8,187,923	4,151,916	1,503,435	1,030,918	1,926,833	1,386,287

	Japanese ↔ English		Russian ↔ English		French ↔ German	
Sentences	10,120,013		12,061,155		7,222,574	
Words	–	274,368,443	182,325,667	210,770,840	145,190,707	123,205,701
Distinct Words	–	2,051,246	2,958,831	2,385,076	1,534,068	2,368,682

	Khmer ↔ English		Pashto ↔ English	
Sentences	4,169,574		1,022,883	
Words	–	77,927,333	14,442,909	13,890,077
Distinct Words	–	1,002,134	365,781	349,261

EU Press Release Parallel Corpus

	Czech ↔ English		German ↔ English		Polish↔ English	
Sentences	452,411		1,631,639		277,984	
Words	7,214,324	7,748,940	26,321,432	27,018,196	6,415,074	6,904,358
Distinct words	141,077	83,733	402,533	197,030	121,451	62,672

Yandex 1M Parallel Corpus

	Russian ↔ English	
Sentences	1,000,000	
Words	24,121,459	26,107,293
Distinct	701,809	387,646

CzEng v2.0 Parallel Corpus

	Czech ↔ English	
Sentences	60,980,645	
Words	757,316,261	848,016,692
Distinct	3,684,081	2,493,804

WikiTitles Parallel Corpus

	Czech ↔ English		German ↔ English		Inuktitut ↔ English		Japanese↔ English	
Sentences	382,336		1,382,681		829		706,012	
Words	916,397	984,247	2,999,545	3,504,013	1213	1213	–	1,867,218
Distinct	206,935	176,156	645,224	547,930	962	938	–	268,391

	Polish ↔ English		Pashto ↔ English		Russian ↔ English	
Sentences	1,006,263		9,869		1,108,789	
Words	2,236,756	2,579,249	20,674	19,519	3,010,302	3,027,765
Distinct	507,571	475,255	9,692	8,899	507,251	434,244

	Tamil↔ English		Chinese↔ English		German↔ French	
Sentences	102,143		836,682		942,017	
Words	237,962	234,380	–	2,267,336	1,989,965	2,363,308
Distinct	72,577	61,267	–	357,440	479,000	423,406

Figure 1: Statistics for the training sets used in the translation task. The number of words and the number of distinct words (case-insensitive) is based on the Moses tokenizer and IndicNLP (https://github.com/anoopkunchukuttan/indic_nlp_library) for Gujarati.

CCMT Corpus

	casia2015	casict2011	casict2015	datum2011	datum2017	neu2017
Sentences	1,050,000	1,936,633	2,036,834	1,000,004	999,985	2,000,000
Words (en)	20,571,578	34,866,598	22,802,353	24,632,984	25,182,185	29,696,442
Distinct words (en)	470,452	627,630	435,010	316,277	312,164	624,420

United Nations Parallel Corpus

	Russian ↔ English		Chinese ↔ English	
Sentences	23,239,280		15,886,041	
Words	570,099,284	601,123,628	–	425,637,920
Distinct	1,446,782	1,027,143	–	769,760

Extra Tamil-English Parallel Data

	PIB		MKB		NLPC	
Sentences	60,836		5,744		8,900	
Words	981,352	1,245,455	91,556	114,415	62,041	75,326
Distinct	96,911	35,954	20,697	9,501	13,794	7,087

	UFAL		Tanzil		PMIndia	
Sentences	169,871		93,540		39,526	
Words	3,335,382	4,537,910	2,595,930	2,822,291	604,814	798,406
Distinct	347,874	70,627	27,711	20,282	70,845	25,074

Extra Japanese-English Parallel Data

	Subtitles		Kyoto		TED	
Sentences	2,801,388		443,849		223,108	
Words	–	23,933,060	–	11,622,252		4,554,409
Distinct	–	161,484	–	191,885	–	60,786

Nunavut Hansard Parallel Corpus

	Inuktitut ↔ English	
Sentences	1,301,736	
Words	10,875,086	20,781,805
Distinct	1,594,280	57,691

Opus Corpus

	Khmer ↔ English		Pashto ↔ English	
Sentences	290,049		123,198	
Words	–	4,537,258	889,520	814,064
Distinct	–	52,496	30,583	20,795

Synthetic parallel data (both directions combined)

	Czech ↔ English		Russian ↔ English		Chinese ↔ English	
Sentences	126,828,081		76,133,209		19,763,867	
Words	2,351,230,606	2,655,779,234	1,511,996,711	1,698,428,744	–	416,567,173
Distinct	5,745,323	3,840,231	5,928,141	3,889,049	–	1,188,933

Wikimatrix Parallel Data

	Czech ↔ English		German ↔ English		Japanese ↔ English		Polish ↔ English	
Sentences	2,094,650		6,227,188		3,895,992		3,085,946	
Words	34,801,119	39,197,172	113,445,806	118,077,685	–	72,320,248	50,061,388	55,736,716
Distinct	1,068,844	798,095	2,855,263	1,827,785		1,106,529	1,312,825	1,096,411

	Russian ↔ English		Tamil ↔ English		Chinese ↔ English		German ↔ French	
Sentences	5,203,872		240,357		2,595,119		3,350,816	
Words	93,828,313	102,937,537	3,057,383	3,766,628	–	58,615,891	68,249,384	59,422,699
Distinct	2,233,043	1,592,190	392,613	262,094	–	1,059,537	1,067,450	1,844,533

Figure 2: Statistics for the training sets used in the translation task. The number of words and the number of distinct words (case-insensitive) is based on the Moses tokenizer and IndicNLP (https://github.com/anoopkunchukuttan/indic_nlp_library) for Tamil.

Monolingual Wikipedia Data

	English	Khmer	Pashto	Tamil
Sentences	67,796,935	132,666	76,557	1,669,257
Words	2,277,495,444	–	3,985,596	22,251,345
Distinct words	8,570,978	–	229,040	1,542,047

News Language Model Data

	English	German	Czech	Russian	Japanese
Sentences	233,501,354	333,313,278	81,708,712	93,827,187	3,446,416
Words	5,578,072,595	6,492,440,544	1,429,535,453	1,702,976,902	–
Distinct words	7,590,931	37,274,673	4,890,810	5,199,379	–

	Polish	Chinese	French	Tamil
Sentences	3,788,276	4,724,008	87,063,385	708,500
Words	66,323,590	–	2,105,883,073	9,421,383
Distinct words	725,050	–	3,736,705	536,423

Document-Split News LM Data (not dedudped)

	Czech	English	German
Sentences	114,101,660	486,139,068	654,097,256
Words	1,798,383,105	10,459,366,947	11,097,364,402
Distinct words	4,765,875	7,857,783	24,538,295

Common Crawl Language Model Data

	English	German	Czech	Russian	Polish
Sent.	3,074,921,453	2,872,785,485	333,498,145	1,168,529,851	1,422,729,881
Words	65,104,585,881	65,147,123,742	6,702,445,552	23,332,529,629	40,639,985,955
Dist.	342,149,665	338,410,238	48,788,665	90,497,177	213,298,869

	Chinese	Inuktitut	Tamil	Pashto	French
Sent.	1,672,324,647	296,730	28,828,239	6,558,180	4,898,012,445
Words	–	1,480,611	632,363,004	218,412,919	126,364,574,036
Dist.	–	448,513	16,780,006	23,531,044	363,878,959

Figure 3: Statistics for the monolingual training sets used in the translation task. The number of words and the number of distinct words (case-insensitive) is based on the Moses tokenizer and IndicNLP (https://github.com/anoopkunchukuttan/indic_nlp_library) for Tamil.

Test Sets

	Czech → EN		EN → Czech		German → EN			EN → German		
Lines.	664		1418		785			1418		
Words	30069	39570	50330	47553	35475	38559	38322	50330	53243	53837
Distinct words	10043	6303	7893	12667	7923	5936	5954	7893	10563	10536

	Inuktitut ↔ EN		Tamil → EN		EN → Tamil		Japanese → EN		EN → Japanese		Khmer ↔ EN	
Lines.	2971		997		1000		993		1000		2320	
Words	36710	68111	15402	19716	25176	19749	–	28446	25176	–	–	45220
Distinct words	14531	5719	6183	3519	4971	8139	–	5195	4971	–	–	5315

	Pashto ↔ EN		Polish → EN		EN → Polish		EN → Russian		German ↔ French	
Lines.	2719		1001		1000		2002		1619	
Words	59245	53754	18472	21852	25176	24346	49862	47909	30422	40180
Distinct words	9071	6305	6685	4274	4971	7997	7772	13042	5428	4727

	Chinese → EN			EN → Chinese			Russia → EN		
Lines.	2000			1418			991		
Words	–	74835	74700	50330	–	–	17249	20346	20704
Distinct words	–	8137	8209	7893	–	–	6328	4091	4066

Figure 4: Statistics for the test sets used in the translation task. In the cases that there are three word counts, these are for source, first target translation, and second target translation. The number of words and the number of distinct words (case-insensitive) is based on the Moses tokenizer and IndicNLP (https://github.com/anoopkunchukuttan/indic_nlp_library) for Tamil

Pair	Domain	Development?	Sentence-split?	Directional?	Documents?
Chinese↔English	news	✗	Only zh→en	✓	✓
Czech↔English	news	✗	✗	✓	✓
French↔German	EU related news	✗	✓	✗	✗
German↔English	news	✗	✗	✓	✓
Inuktitut↔English	news and parliament	✓	✓	✗	Only news
Japanese↔English	news	✓	✓	✓	✓
Khmer↔English	wikipedia	✓	✓	✗	✗
Pashto↔English	wikipedia	✓	✓	✗	✗
Polish↔English	news	✓	✓	✓	✓
Russian↔English	news	✗	✓	✓	✓
Tamil↔English	news	✓	✓	✓	✓

Table 1: The characteristics of the test sets for the news tasks. We show the domain that the test set was drawn from, whether or not we released a development set this year, whether the texts were sentence-split before translation, and whether the direction of translation was preserved. For "directional" test sets, the entire source side of the test set was originally written in the source language, and then translated to the target language. Non-directional test sets are a mixture of "source-original" and "target-original" texts. Finally, we record whether or not the test set contained the original document boundaries.

Language	Documents	Paragraphs	Sentences	Words	Sentences per Paragraph
Czech	102	659	1725	25874	2.62
English	130	1418	2043	44018	1.44
German	118	777	1958	31030	2.52

Table 2: Descriptive statistics of paragraph-split source texts. To count the sentences, we applied the Moses sentence splitter to the texts.

Als Rückzieher sei das aber nicht zu verstehen: "Ganz Ägypten ist der Tahrirplatz".	But that should not be understood as a withdrawal. "All of Egypt is Tahrir square."
"Ich fühle mich unglaublich geehrt und demütig, neben JLo die Latino-Community zu repräsentieren. Denn diese hat eine unglaubliche Stärke in den USA", teilte Shakira in einem Video mit.	"I feel incredibly honored and humbled to be next to J. Lo, representing the Latino community that is such an important force in the United States," Shakira shared in a video.
Man könne die Unternehmen zwar nicht von der Umsatzsteuer auf Sachspenden befreien, erklärte das Ministerium auf eine Frage der Grünen-Bundestagsfraktion, über die die Zeitungen der Funke-Mediengruppe am Freitag berichteten. Die Händler könnten aber den Marktwert der unverkäuflichen Retouren so niedrig ansetzen, dass sie keine oder nur wenig Umsatzsteuer zahlen müssten.	Although it is impossible to exempt companies from VAT on donations in kind, retailers could set the market value of unsaleable returns so low that they would need to pay no or only very little VAT, the Ministry explained in response to a question from the Greens parliamentary group, as reported in newspapers of the Funke media group on Friday.

Table 3: Examples of translations where the translator has split or merged the sentences. The third example is one of the rare examples of a non-trivial merging of the sentence (i.e. there is merging accompanied by reordering)

tionaries (Pavlick et al., 2014), a corpus[3] produced by the University of Moratuwa, the HindEnCorp

[3] https://github.com/nlpcuom/English-Tamil-Parallel-Corpus

Pair	Translator	1-1	%age	Merges	Splits	n-m
Czech→English	A	1573	91.2	26	98	1
English→Czech	A	2013	98.5	10	8	1
German→English	A	1913	97.7	13	19	0
	B	1844	94.2	35	43	0
English→German	A	2017	98.7	9	8	0
	B	1816	88.9	12	203	0

Table 4: How the translators treated sentences when translating the paragraph-split texts. We sentence-split and automatically aligned source and translation. We show the number and percentage of sentences which were translated 1-1, as well as the number of times translators merged or split sentences when translating.

(Kunchukuttan et al., 2018) and English and Tamil wikipedia dumps.

The training corpus for Inuktitut↔English is the recently released Nunavut Hansard Inuktitut–English Parallel Corpus 3.0 (Joanis et al., 2020).

For the Japanese↔English tasks, we added several freely available parallel corpora to the training data. It includes JParaCrawl v2.0 (Morishita et al., 2020), a large web-based parallel corpus, Japanese-English Subtitle Corpus (JESC) (Pryzant et al., 2017), the Kyoto Free Translation Task (KFTT) corpus (Neubig, 2011), constructed from the Kyoto-related Wikipedia articles, and TED Talks (Cettolo et al., 2012).

The monolingual data we provided was similar to last year's, with a 2019 news crawl added to all the news corpora. In addition, we provided versions of the news corpora for Czech, English and German, with both the document and paragraph structure retained. In other words, we did not apply sentence splitting to these corpora, and we retained the document boundaries and text ordering of the originals.

Training, development, and test data for Pashto↔English and Khmer↔English are shared with the Parallel Corpus Filtering Shared Task (Koehn et al., 2020). The training data mostly comes from OPUS (software localization, Tatoeba, Global Voices), the Bible, and special-prepared corpora from TED Talks and the Jehova Witness web site (JW300). The development and test sets were created as part of the Flores initiative (Guzmán et al., 2019) by professional translation of Wikipedia content with careful vetting of the translations. Please refer the to the Parallel Corpus Filtering Shared Task overview paper for details on these corpora.

Some statistics about the training and test materials are given in Figures 1, 2, 3 and 4.

2.3 Submitted Systems

In 2020, we received a total of 153 submissions. The participating institutions are listed in Table 6 and detailed in the rest of this section. Each system did not necessarily appear in all translation tasks. We also included online MT systems (originating from 4 services), which we anonymized as ONLINE-A,B,G,Z.

This year we introduced a new submission tool, OCELoT[4], replacing the matrix that has been used in most previous editions. Using OCELoT gave us more control over the submission and scoring process, for example we were able to limit the number of test submissions by each team, and we also displayed the submissions anonymously to avoid publishing any automatic scores. A screenshot of OCELoT is shown in Figure 5.

For presentation of the results, systems are treated as either *constrained* or *unconstrained*, depending on whether their models were trained only on the provided data. Since we do not know how they were built, the online systems are treated as unconstrained during the automatic and human evaluations.

In the rest of this section, we provide brief details of the submitted systems, for those where the authors provided such details.

2.3.1 AFRL (Gwinnup and Anderson, 2020)

AFRL-SYSCOMB20 is a system combination consisting of two Marian transformer ensembles, one OpenNMT transformer system and a Moses phrase-based system.

AFRL-FINETUNE is an OpenNMT transformer system fine-tuned on newstest2014-2017.

2.3.2 ARIEL XV (Xv, 2020)

ARIEL XV is a Transformer base model trained with the Sockeye sequence modeling toolkit us-

[4] https://github.com/AppraiseDev/OCELoT

English I	ABC News (2), All Africa (5), Brisbane Times (1), CBS LA (1), CBS News (1), CNBC (3), CNN (2), Daily Express (1), Daily Mail (2), Fox News (1), Gateway (1), Guardian (3), Huffington Post (2), London Evening Standard (2), Metro (2), NDTV (7), RTE (7), Reuters (4), STV (2), Seattle Times (3), The Independent (1), The Local (1), The Scotsman (2), The Sun (1), The Telegraph (1), VOA Zimbabwe (1), news.com.au (4),
English II	ABC News (2), Al Jazeera (1), All Africa (6), Brisbane Times (1), CBS LA (1), CNBC (3), CNN (1), Chicago Defender (1), Daily Express (2), Daily Mail (2), Egypt Independent (1), Euronews (1), Guardian (2), Herald Scotland (1), Huffington Post (6), Kazakh TV (1), LA Times (1), London Evening Standard (3), Metro (1), NDTV (6), One India (2), RTE (1), Reuters (1), Russia Today (1), Seattle Times (1), Sky (1), The Independent (1), The Scotsman (4), The Sun (2), UPI (1), news.com.au (3),
English III	ABC News (5), Al Jazeera (3), All Africa (2), Brisbane Times (2), CBS LA (2), CBS News (3), CNBC (5), CNN (6), Chicago Defender (1), Daily Express (2), Daily Mail (2), Euronews (3), Fox News (5), Gateway (1), Guardian (5), Herald Scotland (1), Huffington Post (8), LA Times (2), London Evening Standard (5), Medical Daily (1), Metro (3), NDTV (7), New Republic (1), New York Times (2), Novinite (3), RTE (3), Reuters (8), Russia Today (7), STV (1), Sciencedaily (2), Seattle Times (12), Sky (3), The Independent (2), The Scotsman (1), The Sun (1), The Telegraph (4), UPI (6),
Chinese	China News (64), Chubun (3), Hunan Ribao (5), International Times (10), Jingji Guancha Bao (1), Macao Government (5), Nhan Dan (9), Nikkei (2), Reuters (2), The Australian (2), UN news (2), Xinhua (46), qq.com (1), tsrus.cn (3),
Czech	Aktualne (6), Blesk (13), Denik (7), E15 (3), Hospodářské Noviny (7), Idnes (10), Lidovky (8), Mediafax (3), Neviditelný Pes (2), Novinky (14), Reflex (1), Respekt (5), Týden (9), Česká Pozice (7), České Noviny (7),
German	Allgemeine Zeitung (2), Braunschweiger Zzeitung (3), Dülmener Zeitung (1), Das Bild (2), Der Spiegel (2), Der Standart (2), Deutsche Welle (2), Die Zeit (3), Echo Online (1), Epoch Times (3), Euronews (2), Frankfurter Allgemeine Zeitung (1), Freie Presse (1), Freitag (1), Giessener Anzeiger (1), Goslarsche Zeitung (1), Handelsblatt (2), Heute (2), In Südthüringen (1), Infranken (1), Junge Freiheit (1), Kurier (4), Lübecker Nachrichten (1), Leipziger Volkszeitung (1), Lippische Landes-Zeitung (2), Mittelbayerische Zeitung (2), Mitteldeutsche Zeitung (3), NTV (6), NZZ (5), Nachrichten (2), Neue Osnabrücker Zeitung (1), Neue Presse (1), Neues Deutschland (1), Norddeutsche Neueste Nachrichten (3), OE24 (1), Onetz (1), Passauer Neue Presse (2), Peiner Allgemeine Zeitung (3), Presse Portal (1), Rhein Zeitung (3), Söster Anzeiger (1), Süddeutsche Zeitung (3), Salzburger Nachrichten (4), Schaumburger Nachrichten (1), Schleswig-Holsteinischer Zeitungsverlag (4), Segeberger Zeitung (3), Solinger Tageblatt (1), Stuttgarter Zeitung (1), Tagesspiegel (3), Tiroler Tageszeitung (7), Vaterland (1), Volksblatt (1), Welt (2), Westfälische Nachrichten (1), Westfälischer Anzeiger (1), Wiesbadener Kurier (1), Yahoo (5),
Inuktitut	Nunatsiaq News (36), Nunavut Hansard (1),
Japanese	Fukui Shimbun (6), Hokkaido Shimbun (6), Ise Shimbun (1), Iwaki Minpo (2), Saga Shimbun (3), Sanyo Shimbun (4), Shizuoka Shimbun (15), Ube nippo Shimbun (1), Yahoo (40), Yamagata Shimbun (2),
Polish	Bankier (5), Gazeta Powiatowa (1), Gazeta Prawna (3), Interia (24), Polityka (1), Rzeczpospolita (4), Super Nowosci (3), Sztafeta (1), TVN24 (7), Tygodnik Zamojski (2), WPROST (7), Wyborcza (1), Zycie Podkarpackie (3),
Russian	Argumenti Nedely (6), Argumenty i Fakty (9), BBC Russian (2), Delovoj Peterburg (2), ERR (2), Ekonomika i Zhizn (1), Fakty i Komentarii (3), Gazeta (4), Interfax (3), Izvestiya (7), Kommersant (4), Komsomolskaya Pravda (4), Lenta (7), Moskovskij Komsomolets (3), Nasha Versiya (1), Novye Izvestiya (1), Parlamentskaya Gazeta (5), Rosbalt (5), Rossiskaya Gazeta (1), Russia Today (3), Russkaya Planeta (1), Sport Express (6), Tyumenskaya Oblast Segodnya (1), Vedomosti (2), Vesti (6), Xinhua (2),
Tamil	Aranda Vikatan (11), Dinamalar (2), Makkal Kural (21), One India (21), Viduthalai (15), news.lk (12),

Table 5: Composition of the test sets. English I was used for English to Japanese, Polish, Russian and Tamil, English II was used additionally for English to Russian, and English III (which was not sentence-split) was translated to Czech, German and Chinese. The same document pairs were used in both directions for Inuktitut↔English. For more details see the XML test files. The `docid` tag gives the source and the date for each document in the test set, and the `origlang` tag indicates the original source language.

Team	Institution
AFRL	Air Force Research Laboratory (Gwinnup and Anderson, 2020)
ARIEL XV	Independent submission (Xv, 2020)
CUNI	Charles University (Popel, 2020, 2018; Kocmi, 2020)
DCU	Dublin City University (Parthasarathy et al., 2020)
DEEPMIND	DeepMind (Yu et al., 2020)
DIDI-NLP	DiDi AI Labs (Chen et al., 2020b)
DONG-NMT	(no associated paper)
ENMT	Indepdendent Submission (Kim et al., 2020)
ETRANSLATION	eTranslation (Oravecz et al., 2020)
FACEBOOK AI	Facebook AI (Chen et al., 2020a)
GRONINGEN	University of Groningen (Roest et al., 2020; Dhar et al., 2020)
GTCOM	Global Tone Communication (Bei et al., 2020)
HELSINKINLP	University of Helsinki and Aalto University (Scherrer et al., 2020a)
HUAWEI TSC	Huawei TSC (Wei et al., 2020a)
IIE	Institute of Information Engineering, Chinese Academy of Sciences (Wei et al., 2020b)
MICROSOFT STC INDIA	Microsoft STC India (Goyal et al., 2020)
NICT-KYOTO	NICT-Kyoto (Marie et al., 2020)
NICT-RUI	NICT-Rui (Li et al., 2020)
NIUTRANS	NiuTrans (Zhang et al., 2020)
NRC	National Research Council Canada (Knowles et al., 2020)
OPPO	OPPO (Shi et al., 2020)
PROMT	PROMT (Molchanov, 2020)
SJTU-NICT	SJTU-NICT (Li et al., 2020)
SRPOL	Samsung Research Poland (Krubiński et al., 2020)
TALP UPC	TALP UPC (Escolano et al., 2020)
TENCENT TRANSLATION	Tencent Translation (Wu et al., 2020b)
THUNLP	NLP Lab at Tsinghua University (no associated paper)
TILDE	Tilde (Krišlauks and Pinnis, 2020)
TOHOKU-AIP-NTT	Tohoku-AIP-NTT (Kiyono et al., 2020)
UBIQUS	Ubiqus (Hernandez and Nguyen, 2020)
UEDIN	University of Edinburgh (Bawden et al., 2020a; Germann, 2020)
UEDIN-CUNI	University of Edinburgh and Charles University (Germann et al., 2020)
UQAM_TANLE	Université du Québec à Montréal (no associated paper)
VOLCTRANS	ByteDance AI Lab (Wu et al., 2020a)
WECHAT	WeChat (Meng et al., 2020)
WMTBIOMEDBASELINE	Baseline System from Biomedical Task (Bawden et al., 2020b)
YOLO	American University of Beirut (no associated paper)
ZLABS-NLP	Zoho Corporation (no associated paper)

Table 6: Participants in the shared translation task. Not all teams participated in all language pairs. The translations from the online systems were not submitted by their respective companies but were obtained by us, and are therefore anonymized in a fashion consistent with previous years of the workshop.

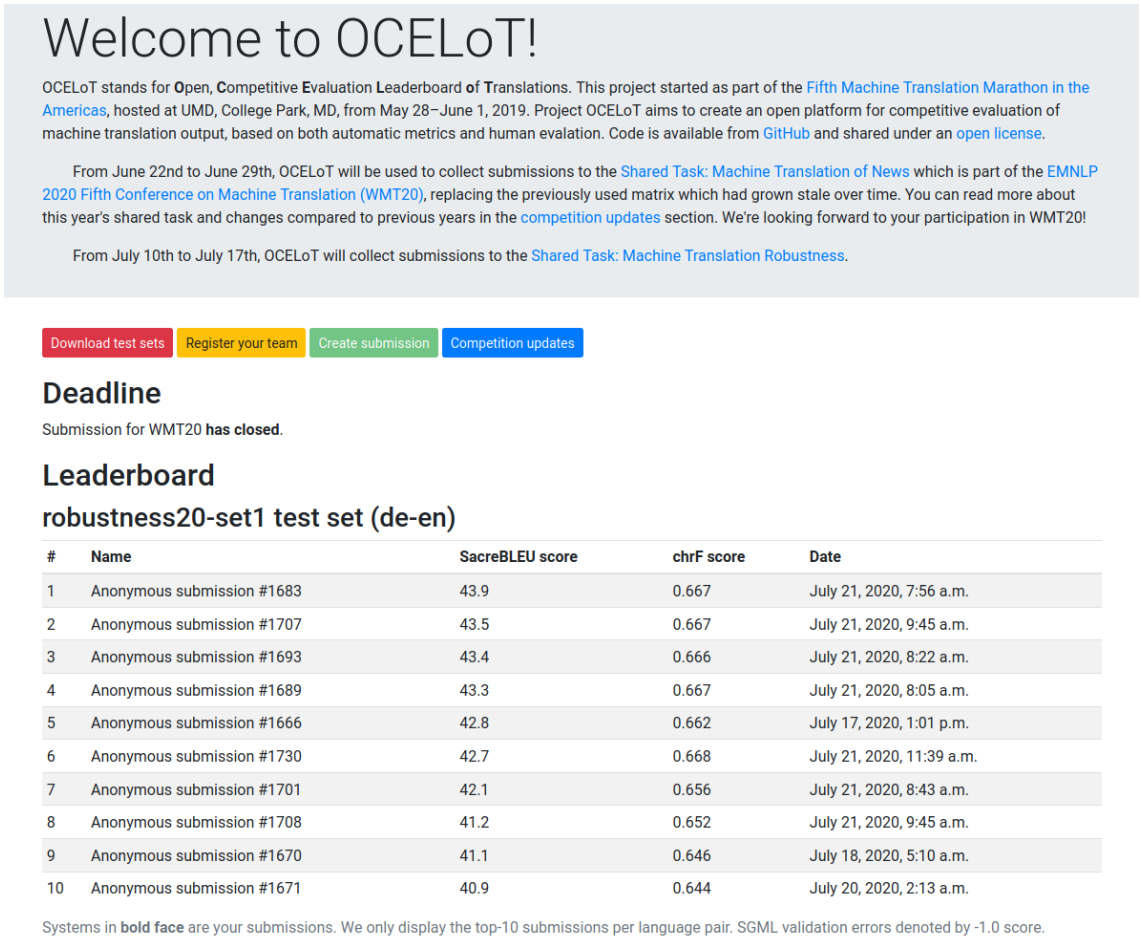


Figure 5: The OCELoT leaderboard tool

ing only the constrained data. The authors experiment with bi-text data filtering, back-translation, rule-based reranking based on translation and language model scores, ensembling several training runs and fine-tuning for sentences similar to the desired domain based on the source side of the test set.

2.3.3 Charles University (CUNI)

CUNI-DocTransformer **(Popel, 2020)** is similar to the sentence-level version (CUNI-T2T-2018, CUBBITT), but trained on sequences with multiple sentences of up to 3000 characters.

CUNI-T2T-2018 **(Popel, 2018),** also called CUBBITT, is exactly the same system as in WMT2018. It is the Transformer model trained according to Popel and Bojar (2018) plus a novel concat-regime backtranslation with checkpoint averaging (Popel et al., 2020), tuned separately for CZ-domain and non CZ-domain articles, possibly handling also translation-direction ("translationese") issues. For cs→en also a coreference preprocessing was used adding the female-gender pronoun where it was pro-dropped in Czech, referring to a human and could not be inferred from a given sentence.

CUNI-Transfer **(Kocmi, 2020)** combines transfer learning from a high-resource language pair Czech–English into the low-resource Inuktitut-English with an additional backtranslation step. Surprising behaviour is noticed when using synthetic data, which can be possibly attributed to a narrow domain of training and test data. The system is the Transformer model in a constrained submission.

CUNI-Transformer **(Popel, 2020)** is similar to the WMT2018 version of CUBBITT, but with 12 encoder layers instead of 6 and trained on CzEng 2.0 instead of CzEng 1.7. The English-Polish version was trained similarly on the provided constrained data.

2.3.4 DCU (Parthasarathy et al., 2020)

DCU participated in the Tamil↔English translations with the Transformer model. Various strate-

gies were tested in order to improve over the baseline, e.g. several techniques of data augmentation and mining as well as a hyperparameter search for better performance of the Transformer model in low-resource scenarios.

2.3.5 DEEPMIND (Yu et al., 2020)

DEEPMIND is a document-level translation system built upon noisy channel factorization. The system optimizes the selection of translations of individual sentences in the document in iterative beam search, replacing sentences with alternative translations. Candidate translations are constructed and later scored using a number of independent components, mainly sequence-to-sequence models trained on large data and highly optimized with techniques of back-translation, distillation, and fine-tuning with in-domain data. MonteCarlo Tree Search decoding and uncertainty estimation are used to improve the robustness of the search for the best sentence translation selection and specialized length models and sentence segmentation help to avoid too short output.

2.3.6 DIDI-NLP (Chen et al., 2020b)

DIDI-NLP is a Transformer model improved with several techniques for model enhancement, including data filtering, data selection, large-scale back-translation, knowledge distillation, fine-tuning, model ensembling, and re-ranking.

Ensembled models include Transformers with relative position attention, larger inner feed-forward network size or reversed source. Multiple domain models based on unsupervised BERT-CLS clusters are used in a dynamically-weighted selection of the next word. The final n-best lists are reranked with MIRA.

2.3.7 DONG-NMT (no associated paper)

No description provided.

2.3.8 ENMT (Kim et al., 2020)

Kim et al. (2020) base their approach on transferring knowledge of domain and linguistic characteristics by pre-training the encoder-decoder model with large amount of in-domain monolingual data through unsupervised and supervised prediction task. The model is then fine-tuned with parallel data and in-domain synthetic data, generated with iterative back-translation. For additional gain, final results are generated with an ensemble model and re-ranked with averaged models and language models.

2.3.9 ETRANSLATION (Oravecz et al., 2020)

ETRANSLATION mainly use the standard training pipeline of Transformer in Marian, using tagged back-translation and other features. Subword units are identified by SentencePiece.

The paper describes the group's concern about computing resources and the practical utility of expensive features like ensembling 2 to 4 bigger models. Techniques that were ineffective in ETRANSLATION's case (e.g. right-to-left model for rescoring English→German or Unicode preprocessing for Japanese→English) are also described.

2.3.10 FACEBOOK AI (Chen et al., 2020a)

FACEBOOK AI focus on low-resource language pairs involving Inuktitut and Tamil using two strategies: (1) exploiting all available data (parallel and monolingual from all languages) and (2) adapting the model to the test domain.

For (1), FACEBOOK AI opt for non-constrained submission, using data derived from CommonCrawl to get strong translation models via iterative backtranslation and self-training and strong language models for noisy channel reranking. Multilingual language models are created using mBART across all the 13 languages of WMT20. For (2), the datasets are tagged for domain, fine-tuned on and further extended with in-domain data.

2.3.11 GRONINGEN

GRONINGEN-ENIU (Roest et al., 2020) investigate the (1) importance of correct morphological segmentation of the polysynthetic Inuktitut, testing rule-based, supervised, semi-supervised as well as unsupervised word segmentation methods, (2) whether or not adding data from a related language (Greenlandic) helps, and (3) whether contextual word embeddings (XLM) improve translation.

GRONINGEN-ENIU use Transformer implemented in Marian with the default setting, improving the performance also with tagged backtranslation, domain-specific data, ensembling and fine-tuning.

GRONINGEN-ENTAM (Dhar et al., 2020) study the effects of various techniques such as linguistically motivated segmentation, back-translation, fine-tuning and word dropout on the English→Tamil News Translation task. Linguis-

tically motivated subword segmentation does not consistently outperform the widely used SentencePiece segmentation despite the agglutinative nature of Tamil morphology. The authors also found that fully-fledged back-translation remains more competitive than its cheaper alternative.

2.3.12 GTCOM (Bei et al., 2020)

GTCOM are unconstrained systems using mBART (Multilingual Bidirectional and Auto-Regressive Transformers), back-translation and forward-translation. Further gains are achieved using rules, language model and RoBERTa model to filter monolingual, parallel sentences and synthetic sentences. The vocabularies are created from both monolingual and parallel data.

2.3.13 HELSINKINLP (Scherrer et al., 2020a)

HELSINKINLP for the Inuktitut-English news translation task focuses on the efficient use of monolingual and related bilingual corpora with multi-task learning as well as an optimized subword segmentation with sampling.

2.3.14 HUAWEI TSC (Wei et al., 2020a)

HUAWEI TSC use Transformer-big with a further increased model size, focussing on standard techniques of careful pre-processing and filtering, back-translation and forward translation, including self-training, i.e. translating one of the sides of the original parallel data. Ensembling of individual training runs is used in the forward as well as backward translation, and single models are created from the ensembles using knowledge distillation. The submission uses THUNMT (Zhang et al., 2017) open-source engine.

2.3.15 IIE (Wei et al., 2020b)

IIE German↔French news translation system is based on the Transformer architecture with some effective improvements. Multiscale collaborative deep architecture, data selection, back translation, knowledge distillation, domain adaptation, model ensemble and re-ranking are employed and proven effective in the experiments.

2.3.16 MICROSOFT STC INDIA (Goyal et al., 2020)

Focusing on English↔Tamil, MICROSOFT STC INDIA experiment with "contact relatedness" of languages, i.e. using Hindi-English data in joint training. Hindi texts first have to be mapped from the Devanagari script to Tamil characters in a lossy but deterministic way. Further gains are obtained from tagged back-translation and other variants of back-translation are also examined (noisification or back-translating with right-to-left models).

Transformer implemented in fairseq is used, with smaller than "base" models due to limited training data.

2.3.17 NICT-KYOTO (Marie et al., 2020)

NICT-KYOTO is a combination of neural machine translation systems processed through n-best list reranking. The systems combined are Transformer-based trained with Marian and Fairseq with and without using tagged back-translation. All the systems are constrained, and the final primary submission is selected on the basis of the BLEU score obtained on the official validation data.

2.3.18 NICT-RUI (Li et al., 2020)

NICT-RUI (Li et al., 2020) NICT-RUI is closely related to SJTU-NICT using large XLM model to improve NMT but the exact relation is unclear.

2.3.19 NIUTRANS (Zhang et al., 2020)

NIUTRANS gain their performance from focussed attention to six areas: (1) careful data preprocessing and filtering, (2) iterative back-translation to generate additional training data, (3) using different model architectures, such as wider and/or deeper models, relative position representation and relative length, to enhance the diversity of translations, (4) iterative knowledge distillation by in-domain monolingual data, (5) iterative fine-tuning for domain adaptation using small training batches, (6) rule-based post-processing of numbers, names and punctuation.

For low-resource language pairs, multi-lingual seed models are used.

2.3.20 NRC (Knowles et al., 2020)

The NRC systems are hybrids of Transformer models trained with Sockeye, with one ensembled system for news domain translation and one for Hansard domain translation. Data was pre-processed with language-specific punctuation and character preprocessing, tokenization, and BPE. They were trained with domain tagging, domain-specific finetuning, ensembles of 3 systems per domain, BPE-dropout (EN-IU), and tagged back-translation (IU-EN).

2.3.21 OPPO (Shi et al., 2020)

OPPO train Marian for some language pairs and fairseq for others, relying on a number of mature techniques including careful corpus filtering, iterative forward and backward translation, fine-tuning on the original parallel data, ensembling of several different models, and complex reranking which uses forward (source-to-target) scorers, backward scorers (target-to-source) and language models (monolingual), each group again building upon ensembles and being applied left-to-right as well as right-to-left.

Each language pair received targeted attention, discussing training data properties, varying the process as needed and choosing from several possible final models.

2.3.22 PROMT (Molchanov, 2020)

PROMT BASELINE TRANSFORMER uses MarianNMT, shared vocabulary, 16k BPE merge operations and it is trained on unconstrained data.

PROMT BASIC TRANSFORMER uses separate vocabs (16k source + 32k target), and tied embeddings.

PROMT MULTILINGUAL 4-TO-EN is a multilingual system trained to translate from Croatian, Serbian, Slovak and Czech to English. It is a basic Transformer configuration with shared vocabulary.

PROMT MULTILINGUAL PL-EN is a Polish↔English system trained jointly in both directions. It uses basic Transformer configuration and shared vocabulary.

None of PROMT systems are constrained.

2.3.23 SJTU-NICT (Li et al., 2020)

SJTU-NICT represents two different main approaches. For News Translation Task, (1) cross-lingual language models (XLM) are used in an additional encoder to benefit from language-independent sentence representations from both the source and target side for Polish→English. For English→Chinese, which includes document-level information, three-stage training is used to train Longformer (Transformer with attention extended to the full document).

2.3.24 SRPOL (Krubiński et al., 2020)

No short description provided.

2.3.25 TALP UPC (Escolano et al., 2020)

No short description provided.

2.3.26 TENCENT TRANSLATION (Wu et al., 2020b)

No short description provided.

2.3.27 THUNLP (no associated paper)

No description provided.

2.3.28 TILDE (Krišlauks and Pinnis, 2020)

For WMT 2020, Tilde developed English↔Polish (separate constrained and unconstrained submissions) and Polish↔English (constrained only) NMT systems. Tilde experimented with morpheme splitting prior to byte-pair encoding, dual conditional cross-entropy filtering, sampling-based backtranslation of source-domain-adherent monolingual data, and right-to-left reranking. The submitted translations were produced using ensembles of Transformer base and Transformer big models, which were trained using back-translated data, and right-to-left re-ranking.

2.3.29 TOHOKU-AIP-NTT (Kiyono et al., 2020)

TOHOKU-AIP-NTT used Transformer-based Encoder-Decoder model with 8 layers and feed forward dimension of 8192. Synthetic data were created via beam back-translation from monolingual data available for each language and incorporated to the training using tagged backtranslation. The bitext was oversampled so that the model saw the bitext and synthetic data in 1:1 ratio. After training, the model was finetuned with newstest corpus.

An ensemble of four models was used to generate candidate translation, which were in turn re-ranked using scores from following components: (1) source-to-target right-to-left model, (2) target-to-source left-to-right model, (3) target-to-source right-to-left model, (4) masked language model (RoBERTa), and (5) uni-directional language model (Transformer-LM).

2.3.30 MULTILINGUAL-UBIQUS (Hernandez and Nguyen, 2020)

UBIQUS performed a single submission, based on an unconstrained multilingual setup. The approach consists of jointly training a traditional Transformer model on several agglutinative languages in order to benefit from them for the low-resource English-Inuktitut task. For that purpose,

the dataset was extended with other linguistically near languages (Finnish, Estonian), as well as in-house datasets introducing more diversity to the domain.

2.3.31 UEDIN

UEDIN **(Bawden et al., 2020a)** for the very low-resource English-Tamil involved exploring pretraining, using both language model objectives and translation using an unrelated high-resource language pair (German-English), and iterative backtranslation. For English-Inuktitut, UEDIN explored the use of multilingual systems.

UEDIN-DEEN **and** UEDIN-ENDE **(Germann, 2020)** ensemble big transformer models trained in three stages: First, base transformer models were trained on available high-quality parallel data. These models were used to rank and select parallel data from crawled and automatically matched parallel data (Paracrawl, Commoncrawl, etc.). 2nd-generation big transformers were then trained on the combined parallel data. These models were used for back-translation. Original and back-translated data was then used to the final 3rd-generation models.

2.3.32 UEDIN-CUNI (Germann et al., 2020)

UEDIN-CUNI CSEN STUDENT **and** UEDIN-CUNI ENCS STUDENT are compact, efficient student models that distill knowledge from larger teacher models. All models are variants of the transformer architecture. The teacher models were used to translate the source side of the training data to create synthetic training data for the student models.

2.3.33 UQAM_TANLE

No description provided.

2.3.34 VOLCTRANS (Wu et al., 2020a)

VOLCTRANS aims at building a general training framework which can be well applied to different translation. directions. Techniques used in the submitted systems include optional multilingual pre-training (mRASP) for low resource languages, very deep Transformer or dynamic convolution models up to 50 encoder layers, iterative back-translation, knowledge distillation, model ensemble and development set fine-tuning. The key ingredient of the process seems the strong focus on diversification of the (synthetic) training data, using multiple scalings of the Transformer model and dynamic convolution, random upsamplings of the parallel data, creation of multiple back-translated corpus variants or random ensembling which uses not a fixed set of ensembled models but rather a random checkpoint of each of them.

2.3.35 WECHAT (Meng et al., 2020)

WECHAT is based on the Transformer with effective variants and the DTMT architecture. The experiments include data selection, several synthetic data generation approaches (i.e., back-translation, knowledge distillation, and iterative in-domain knowledge transfer), advanced finetuning approaches and self-bleu based model ensemble.

2.3.36 WMTBIOMEDBASELINE (Bawden et al., 2020b)

WMTBIOMEDBASELINE are the baseline systems from the Biomedical Translation Task.

2.3.37 YOLO (no associated paper)

No description provided.

2.3.38 ZLABS-NLP

ZLABS-NLP used SentencePiece for subword segmentation, otherwise the model including hyper-parameters is the same as described by Ott et al. (2018) and implemented in FairSeq. Probably, OpenNMT-py was used during training (back-translation for Tamil).

3 Human Evaluation

A human evaluation campaign is run each year to assess translation quality and to determine the official ranking of systems taking part in the news translation task. This section describes how data for the human evaluation is prepared, the process of collecting human assessments, and computation of the official results of the shared task.

3.1 Direct Assessment

Since running a comparison of *direct assessments* (DA, Graham et al., 2013, 2014, 2016) and relative ranking in 2016 (Bojar et al., 2016) and verifying a high correlation of system rankings for the two methods, as well as the advantages of DA, such as quality controlled crowd-sourcing and linear growth relative to numbers of submissions, we have employed DA as the primary mechanism for evaluating systems. With DA human evaluation,

human assessors are asked to rate a given translation by how adequately it expresses the meaning of the corresponding reference translation or source language input on an analogue scale, which corresponds to an underlying absolute 0–100 rating scale.[5] No sentence or document length restriction is applied during manual evaluation. Direct Assessment is also employed for evaluation of video captioning systems at TRECvid (Graham et al., 2018; Awad et al., 2019) and multilingual surface realisation (Mille et al., 2018, 2019).

3.1.1 Source and Reference-based Evaluations

The earlier DA evaluations that we performed were all referenced based, as described above, however in 2018 we trialled source-based evaluation for the first time, in English to Czech translation. In this configuration, the human assessor is shown the source input and system output only (with no reference translation shown). This approach has the advantage of freeing up the human-generated reference translation so that it can be included in the evaluation to provide an estimate of human performance. As was the approach in WMT19, since we would like to restrict human assessors to only evaluate translation *into* their native language, we again restrict bilingual/source-based evaluation to evaluation of translation for out-of-English language pairs. This is especially relevant since we have a large group of volunteer human assessors with native language fluency in non-English languages and high fluency in English, while we generally lack the reverse, i.e. native English speakers with high fluency in non-English languages.

3.1.2 Translationese

Prior to WMT19, all the test sets included a mix of sentence pairs that were originally in the source language, and then translated to the target language, and sentence pairs that were originally in the target language but translated to the source language. The inclusion of the latter "reverse-created" sentence pairs has been shown to introduce biases into the evaluations, particularly in terms of BLEU scores (Graham et al., 2020), so we avoid it where possible. As detailed in Section 2, most of our test sets do not include reverse-created sentence pairs, except when there were resource constraints on the creation of the test sets.

3.1.3 Document Context

Prior to WMT19, the issue of including document context was raised within the community (Läubli et al., 2018; Toral et al., 2018) and at WMT19 a range of DA styles were subsequently tested that included document context. In WMT19, two options were run, firstly, an evaluation that included the document context "+DC" (with document context), and secondly, a variation that omitted document context "−DC" (without document context). This year, for language pairs for which document context was available in the test set, we therefore include this context when evaluating translations for systems. Although we include document context, ratings are nevertheless collected on the segment-level, motivated by the power analysis described in Graham et al. (2019) and Graham et al. (2020). The particular details on how document context is made available to assessors depends on the translation direction, as described in more detail in Sections 3.2 and 3.3 below for translation into English and out of English, resp.

In the following, we use the following abbreviations to describe annotation style: SR+DC for translation direction where assessors rank individual segments (Segment Ranking, SR) and have access to the full document, SR−DC for translation directions where document context is not available and assessors see individual sentences in random order.

Fully document-level evaluation (DR+DC, document-level ranking with document context available) as trialled last year where we asked for a single score given the whole document is problematic in terms of statistical power and inconclusive ties, as shown in Graham et al. (2019); Graham et al. (2020), and we subsequently did not include this approach for any into-English language this year.

As in previous years, the SR−DC annotation is organized into "HITs" (following the Mechanical Turk's term "human intelligence task"), each containing 100 screens.

[5] Past work has investigated the degree to which employment of a reference translation in DA evaluations could introduce bias into evaluation results and showed no significant evidence of reference-bias (Ma et al., 2017).

	Seg Rating + Doc Context (SR+DC)	Seg Rating − Doc Context (SR−DC)
Chinese to English	**M**	
Czech to English	**M**	
German to English	**M**	
Inuktitut to English		**M**
Khmer to English		**M**
Japanese to English	**M**	
Pashto to English		**M**
Polish to English	**M**	
Russian to English	**M**	
Tamil to English	**M**	

Table 7: Summary of human evaluation configurations for monolingual translation for into-English language pairs; M denotes reference-based/monolingual human evaluation in which the machine translation output was compared to human-generated reference

Language Pair	**Sys.**	**Assess.**	**Assess/Sys**
Czech→English	12	10,703	891.9
German→English	13	14,303	1,100.2
Inuktitut→English	11	13,897	1,263.4
Japanese→English	10	11,234	1,123.4
Khmer→English	7	7,944	1,134.9
Polish→English	14	14,146	1,010.4
Pashto→English	6	8,162	1,360.3
Russian→English	12	12,783	1,065.2
Tamil→English	14	8,899	635.6
Chinese→English	17	34,596	2,035.1
Total to-English	116	136,667	1,178.2

Table 8: Amount of data collected in the WMT20 manual evaluation campaign for evaluation into-English; after removal of quality control items.

3.2 Human Evaluation of Translation into-English

A summary of the human evaluation configurations run this year in the news task for into-English language pairs is provided in Table 7.

In terms of the News translation task manual evaluation for into-English language pairs, a total of 654 turker accounts were involved.[6] 654,583 translation assessment scores were submitted in total by the crowd, of which 166,868 were provided by workers who passed quality control.

System rankings are produced from a large set of human assessments of translations, each of which indicates the absolute quality of the output of a system. Table 8 shows total numbers of human assessments collected in WMT20 for into-English language pairs contributing to final scores for systems.[7]

[6]Numbers do not include the 2,233 workers on Mechanical Turk who did not pass quality control.

[7]Number of systems for WMT20 includes three "human" systems comprising human-generated reference translations used to provide human performance estimates.

3.2.1 Crowd Quality Control

We run two configurations of DA, one with document context, segment-rating with document context (SR+DC), for languages for which this information was available and one without document context, for the remainder, segment rating without document context (SR-DC). We describe quality control details and both methods of ranking systems for into-English language pairs in detail below.

Standard DA HIT Structure (SR−DC) In the standard DA HIT structure (without document context), three kinds of quality control translation pairs are employed as described in Table 9: we repeat pairs expecting a similar judgment (Repeat Pairs), damage MT outputs expecting significantly worse scores (Bad Reference Pairs) and use references instead of MT outputs expecting high scores (Good Reference Pairs). For each of these three types, we include the MT output, along with its corresponding control.

In total, 60 items in a 100-translation HIT serve in quality control checks but 40 of those are regular judgments of MT system outputs (we exclude assessments of bad references and ordinary reference translations when calculating final scores). The effort wasted for the sake of quality control is thus 20%.

Also in the standard DA HIT structure, within each 100-translation HIT, the same proportion of translations are included from each participating system for that language pair. This ensures the final dataset for a given language pair contains roughly equivalent numbers of assessments for each participating system. This serves three purposes for making the evaluation fair. Firstly, for the point estimates used to rank systems to be reliable, a sufficient sample size is needed and the most efficient way to reach a sufficient sample size for all systems is to keep total numbers of judgments roughly equal as more and more judgments are collected. Secondly, it helps to make the evaluation fair because each system will suffer or benefit equally from an overly lenient/harsh human judge. Thirdly, despite DA judgments being absolute, it is known that judges "calibrate" the way they use the scale depending on the general observed translation quality. With each HIT including all participating systems, this effect is

Repeat Pairs:	Original System output (10)	An exact repeat of it (10);
Bad Reference Pairs:	Original System output (10)	A degraded version of it (10);
Good Reference Pairs:	Original System output (10)	Its corresponding reference translation (10).

Table 9: Standard DA HIT structure quality control translation pairs hidden within 100-translation HITs, numbers of items are provided in parentheses.

averaged out. Furthermore apart from quality control items, HITs are constructed using translations sampled from the entire set of outputs for a given language pair.

Document-Level DA HIT Structure (SR+DC) Collection of segment-level ratings with document context (Segment Rating + Document Context) involved constructing HITs so that each sentence belonging to a given document (produced by a single MT system) was displayed to and rated in turn by the human annotator.

Quality control items for this set-up was carried out as follows with the aim of constructing a HIT with as close as possible to 100 segments in total:

1. All documents produced by all systems are pooled;[8]

2. Documents are then sampled at random (without replacement) and assigned to the current HIT until the current HIT comprises no more than 70 segments in total;

3. Once documents amounting to close to 70 segments have been assigned to the current HIT, we select a subset of these documents to be paired with quality control documents; this subset is selected by repeatedly checking if the addition of the number of the segments belonging to a given document (as quality control items) will keep the total number of segments in the HIT below 100; if this is the case it is included; otherwise it is skipped until the addition of all documents has been checked. In doing this, the HIT is structured to bring the total number of segments as close as possible to 100 segments in total within a HIT but without selecting documents in any systematic way such as selecting them based on fewest segments, for example.

4. Once we have selected a core set of original system output documents and a subset of them to be paired with quality control versions for each HIT, quality control documents are automatically constructed by altering the sentences of a given document into a mixture of three kinds of quality control items used in the original DA segment-level quality control: bad reference translations, reference translations and exact repeats (see below for details of bad reference generation);

5. Finally, the documents belonging to a HIT are shuffled.

Construction of Bad References As in previous years, bad reference pairs were created automatically by replacing a phrase within a given translation with a phrase of the same length, randomly selected from n-grams extracted from the full test set of reference translations belonging to that language pair. This means that the replacement phrase will itself comprise a mostly fluent sequence of words (making it difficult to tell that the sentence is low quality without reading the entire sentence) while at the same time making its presence highly likely to sufficiently change the meaning of the MT output so that it causes a noticeable degradation. The length of the phrase to be replaced is determined by the number of words in the original translation, as follows:

Translation Length (N)	# Words Replaced in Translation
1	1
2–5	2
6–8	3
9–15	4
16–20	5
>20	$\lfloor$ N/4 $\rfloor$

3.2.2 Annotator Agreement

When an analogue scale (or 0–100 point scale, in practice) is employed, agreement cannot be measured using the conventional Kappa coefficient, ordinarily applied to human assessment when judgments are discrete categories or preferences. Instead, to measure consistency we fil-

[8] If a "human" system is included to provide a human performance estimate, it is also considered a system during quality control set-up.

ter crowd-sourced human assessors by how consistently they rate translations of known distinct quality using the bad reference pairs described previously. Quality filtering via bad reference pairs is especially important for the crowd-sourced portion of the manual evaluation. Due to the anonymous nature of crowd-sourcing, when collecting assessments of translations, it is likely to encounter workers who attempt to game the service, as well as submission of inconsistent evaluations and even robotic ones. We therefore employ DA's quality control mechanism to filter out low quality data, facilitated by the use of DA's analogue rating scale.

Assessments belonging to a given crowd-sourced worker who has not demonstrated that he/she can reliably score bad reference translations significantly lower than corresponding genuine system output translations are filtered out. A paired significance test is applied to test if degraded translations are consistently scored lower than their original counterparts and the p-value produced by this test is used as an estimate of human assessor reliability. Assessments of workers whose p-value does not fall below the conventional 0.05 threshold are omitted from the evaluation of systems, since they do not reliably score degraded translations lower than corresponding MT output translations.

Table 10 shows the number of workers participating in the into-English translation evaluation who met our filtering requirement in WMT20 by showing a significantly lower score for bad reference items compared to corresponding MT outputs, and the proportion of those who simultaneously showed no significant difference in scores they gave to pairs of identical translations. We removed data from the non-reliable workers in all language pairs.

3.3 Human Evaluation of Translation out-of-English

Human evaluation of out-of-English translations features a bilingual/source-based evaluation campaign that enlists the help of participants in the shared task. As usual, each team was asked to contribute around 8 hours annotation time, which we estimated at 16 HITs per each primary system submitted, with each HIT including 100 segment translations. Unfortunately, not all participating teams were able to provide requested number of assessments, hence, to collect the required number of assessments per MT system, we also employed external translators in a separate campaign. The contracted translators contributed with one third of total number of assessments. Both campaigns utilized document-level DA and were run for all out-of-English language pairs, which test sets include document-level segmentation.

For English→Khmer, English→Pashto, French→German, and German→French, whose test sets do not provide document boundaries, segment-level DA evaluation without document context (SR–DC) was performed, enlisting the effort of translators.

For English→Inuktitut, since we expected no participants to speak Inuktitut, the NRC hired native speakers through the Pirurvik Centre to conduct most of the DA evaluation. Due to the delays in starting the evaluation campaign, they were only able to complete the evaluation a few days before the conference, and could only annotate the news half of the test set. The Hansard half of the test set was not assessed in time for this report, but plans are being made to continue the evaluation after the conference. Updated rankings should be provided at a future date.

In terms of the News translation task document-level manual evaluation for out-of-English language pairs, a total of 1,189 researcher/translator accounts were involved, and 248,597 translation assessment scores were contributed in total (with quality control pairs), including 18,108 document ratings. For the segment-level campaigns (i.e. English→Khmer, English→Pashto, German→French and French→German) we had 300 accounts and 65872 scores collected in total. Statistics per language pair are summarized in Table 11. For data collection we again used the open-source Appraise[9] (Federmann, 2012). The effort that goes into the manual evaluation campaign each year is impressive, and we are grateful to all participating individuals and teams for their work.

3.3.1 Document-Level Assessment

This year's human evaluation for out-of-English language pairs features an improved document-level direct assessment configuration that extends the context span to entire documents for a more reliable machine translation evaluation (Castilho

[9] https://github.com/AppraiseDev/Appraise

		All	(A) Sig. Diff. Bad Ref.	(A) & No Sig. Diff. Exact Rep.
SR−DC	Inuktitut→English	464	87 (19%)	81 (93%)
	Khmer→English	529	60 (11%)	56 (93%)
	Pashto→English	321	46 (14%)	46 (100%)
	Total	**1,126**	**169 (15%)**	**158 (93%)**
SR+DC	Czech→English	247	50 (20%)	43 (86%)
	German→English	343	84 (24%)	77 (92%)
	Japanese→English	422	81 (19%)	74 (91%)
	Polish→English	367	87 (24%)	77 (89%)
	Russian→English	360	109 (30%)	89 (82%)
	Tamil→English	235	71 (30%)	65 (92%)
	Chinese→English	878	178 (20%)	158 (89%)
	Total	**1,804**	**482 (27%)**	**423 (88%)**
	Overall	**2,930**	**651 (22%)**	**581 (90%)**

Table 10: Number of crowd-sourced workers taking part in the reference-based SR−DC campaign; (A) those whose scores for bad reference items were significantly lower than corresponding MT outputs; those of (A) whose scores also showed no significant difference for exact repeats of the same translation.

Language Pair	Sys.	Assess.	Assess/Sys
English→Czech	13	37,535	2,887.3
English→German	17	19,102	1,123.6
English→Inuktitut	12	21,816	1,818.0
English→Japanese	12	24,341	2,028.4
English→Polish	15	20,162	1,344.1
English→Russian	10	21,618	2,161.8
English→Tamil	16	10,123	632.7
English→Chinese	14	46,207	3,300.5
Total document-level	109	200,904	1,843.2
German→French	7	14,470	2067.1
French→German	9	16,844	1871.6
English→Khmer	8	13,393	1,674.1
English→Pashto	7	13,267	1,895.3
Total segment-level	31	57,974	1,870.1

Table 11: Amount of data collected in the WMT20 manual document- and segment-level evaluation campaigns for bilingual/source-based evaluation out of English and non-English pairs.

et al., 2020; Laubli et al., 2020). It differs from SR+DC DA introduced in WMT19 (Bojar et al., 2019), and still used in into-English human evaluation this year, where a single segment from a document is provided on a screen at a time, followed by showing the entire document during annotation. Figure 6 shows a screenshot of the document-level direct assessment interface introduced this year.[10] Annotators see the entire document on a screen. In the default scenario, an annotator scores individual segments one-by-one and, after scoring all of them, on the same screen, the annotator then judges the translation of the entire document displayed. Annotators can, however, revisit and update scores of previously assessed segments at any point of the annotation of the given document.

3.3.2 Quality Control

For the document-level evaluation of out-of-English translations, HITs were generated using the same method as described for the SR+DC evaluation of into-English translations in Section 3.2.1 with minor modifications. Source-based DA allows to include human references in the evaluation as another system to provide an estimate of human performance. Human references were added to the pull of system outputs prior to sampling documents for tasks generation. If multiple references are available, which is the case for English→German (3 alternative reference translations, including 1 generated using the paraphrasing method of Freitag et al. (2020)) and English→Chinese (2 translations), each reference is assessed individually.

Since the annotations are made by researchers and professional translators who ensure a bet-

[10] Compare with Figures 3 and 4 in Bojar et al. (2019).

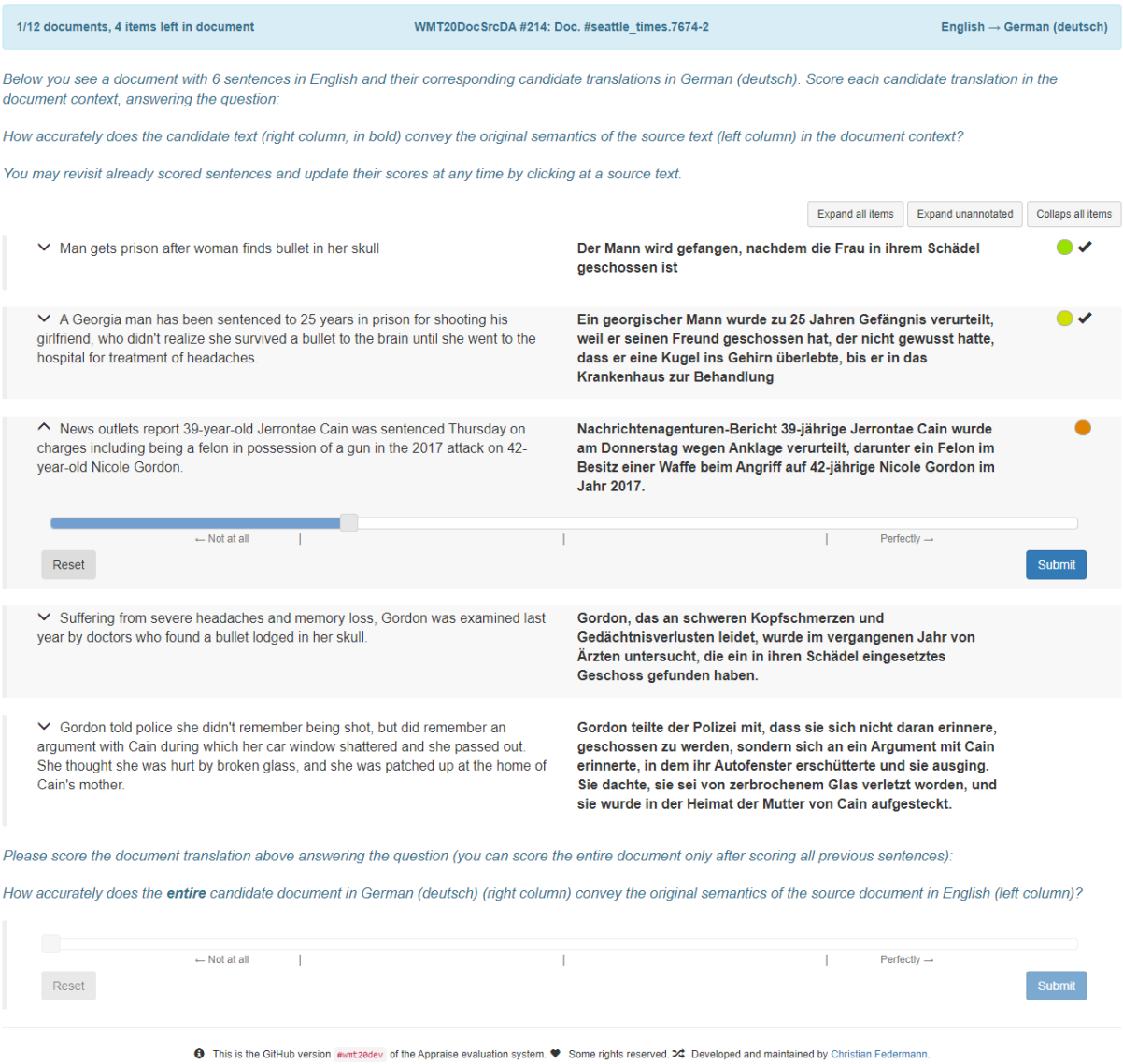


Figure 6: Screen shot of the new document-level DA configuration in the Appraise interface for an example assessment from the human evaluation campaign. The annotator is presented with the entire translated document randomly selected from competing systems (anonymized) and is asked to rate the translation of individual segments and the entire document on sliding scales.

ter quality of assessments than the crowd-sourced workers, only bad references are used as quality control items. Instead of sampling initial documents with close to 70 segments, we sample documents with 88 segments, and then a subset of documents with around 12 segments is selected to be converted into bad references. The remaining of the HIT creation process remains the same.

3.4 Producing the Human Ranking

In all set-ups, similar to previous years, system rankings were arrived at in the following way. Firstly, in order to iron out differences in scoring strategies of distinct human assessors, human assessment scores for translations were first standardized according to each individual human assessor's overall mean and standard deviation score. This year all rankings for to-English translation were arrived at via segment ratings (SR−DC, SR+DC), average standardized scores for individual segments belonging to a given system were then computed, before the final overall DA score for a given system is computed as the average of its segment scores (Ave z in Table 12). Results are also reported for average scores for systems, computed in the same way but without any score standardization applied (Ave % in Table 12).

Table 13 shows official news task results for translation out of English, where lines indicate

Chinese→English

Ave.	Ave. z	System
77.5	0.102	VolcTrans
77.6	0.089	DiDi-NLP
77.4	0.077	WeChat-AI
76.7	0.063	Tencent-Translation
77.8	0.060	Online-B
78.0	0.051	DeepMind
77.5	0.051	OPPO
76.5	0.028	THUNLP
76.0	0.016	SJTU-NICT
72.4	0.000	Huawei-TSC
76.1	−0.017	Online-A
74.8	−0.029	HUMAN
71.7	−0.071	Online-G
74.7	−0.078	dong-nmt
72.2	−0.106	zlabs-nlp
72.6	−0.135	Online-Z
67.3	−0.333	WMTBiomedBaseline

Czech→English

Ave.	Ave. z	System
78.3	0.118	CUNI-DocTransformer
77.5	0.071	OPPO
74.8	0.041	Online-B
75.3	0.034	CUNI-Transformer
73.8	0.018	Online-A
73.7	−0.037	SRPOL
74.1	−0.049	UEDIN-CUNI
74.1	−0.065	CUNI-T2T-2018
72.5	−0.069	Online-G
71.8	−0.080	Online-Z
71.9	−0.094	PROMT-NMT
72.0	−0.141	zlabs-nlp

German→English

Ave.	Ave. z	System
82.6	0.228	VolcTrans
84.6	0.220	OPPO
82.2	0.186	HUMAN
81.5	0.179	Tohoku-AIP-NTT
81.3	0.179	Online-A
81.5	0.172	Online-G
79.8	0.171	PROMT-NMT
82.1	0.167	Online-B
78.5	0.131	UEDIN
78.8	0.085	Online-Z
74.2	−0.079	WMTBiomedBaseline
71.1	−0.106	zlabs-nlp
20.5	−1.618	yolo

Inuktitut→English

Ave.	Ave. z	System
73.1	0.168	NiuTrans
72.9	0.167	Facebook-AI
71.2	0.100	CUNI-Transfer
70.7	0.096	Groningen
70.3	0.072	SRPOL
71.1	0.066	Helsinki
70.2	0.055	NRC
70.2	0.054	UEDIN
70.1	0.047	UQAM-TanLe
68.8	0.006	NICT-Kyoto
68.4	−0.035	OPPO

Japanese→English

Ave.	Ave. z	System
75.1	0.184	Tohoku-AIP-NTT
76.4	0.147	NiuTrans
74.1	0.088	OPPO
75.2	0.084	NICT-Kyoto
73.3	0.068	Online-B
70.9	0.026	Online-A
71.1	0.019	eTranslation
64.1	−0.208	zlabs-nlp
66.0	−0.220	Online-G
61.7	−0.240	Online-Z

Khmer→English

Ave.	Ave. z	System
69.0	0.168	Online-B
69.4	0.146	GTCOM
68.5	0.136	Huawei-TSC
62.6	−0.047	VolcTrans
58.1	−0.210	OPPO
56.9	−0.222	Online-Z
55.5	−0.282	Online-G

Pashto→English

Ave.	Ave. z	System
67.3	0.032	Online-B
66.7	0.024	GTCOM
65.5	−0.016	Huawei-TSC
62.7	−0.106	VolcTrans
62.1	−0.164	OPPO
61.0	−0.195	Online-Z

Polish→English

Ave.	Ave. z	System
77.2	0.131	SRPOL
76.7	0.097	Online-G
77.7	0.096	NICT-Rui
77.9	0.094	Online-B
78.1	0.085	SJTU-NICT
76.6	0.083	Online-A
75.2	0.050	OPPO
77.3	0.006	Online-Z
78.1	−0.003	CUNI-Transformer
76.1	−0.038	NICT-Kyoto
73.3	−0.041	VolcTrans
73.2	−0.048	PROMT-NMT
74.3	−0.072	Tilde
74.0	−0.130	zlabs-nlp

Russian→English

Ave.	Ave. z	System
79.3	0.124	Online-G
80.9	0.114	Online-A
79.7	0.113	OPPO
80.6	0.104	eTranslation
79.5	0.096	PROMT-NMT
80.2	0.072	Online-B
79.9	0.062	HUMAN
77.7	0.042	ariel xv
79.2	0.026	AFRL
76.0	−0.016	DiDi-NLP
75.2	−0.022	Online-Z
71.7	−0.153	zlabs-nlp

Tamil→English

Ave.	Ave. z	System
68.7	0.203	GTCOM
70.3	0.202	OPPO
68.9	0.176	Online-B
73.9	0.173	Facebook-AI
70.9	0.150	NiuTrans
71.9	0.116	VolcTrans
64.5	0.007	Online-Z
66.4	0.001	zlabs-nlp
67.5	−0.016	Microsoft-STC-India
60.8	−0.020	UEDIN
64.5	−0.068	Online-A
63.4	−0.078	DCU
53.7	−0.398	Online-G
53.9	−0.451	TALP-UPC

Table 12: Official results of WMT20 News Translation Task for translation into-English. Systems ordered by DA score z-score; systems within a cluster are considered tied; lines indicate clusters according to Wilcoxon rank-sum test $p < 0.05$; grayed entry indicates resources that fall outside the constraints provided.

clusters according to Wilcoxon rank-sum test $p < 0.05$. For evaluation of English→Inuktitut insufficient data resulted in a small sample size of human assessments per system and as a result some systems that fall within the same cluster are likely to do so simply due to low statistical power (Graham et al., 2020).

Human performance estimates arrived at by evaluation of human-produced reference translations are denoted by "HUMAN" in all tables. Note that "HUMAN-P" is a human-produced paraphrase of HUMAN-A, according to the method proposed by Freitag et al. (2020). Clusters are identified by grouping systems together according to which systems significantly outperform all others in lower ranking clusters, according to Wilcoxon rank-sum test.

Appendix A shows the underlying head-to-head significance test official results for all pairs of systems. All data collected during the human evaluation is available at `http://www.statmt.org/wmt20/results.html`.

In terms of human and machine quality comparisons in results, it is clear from the source-based evaluation of English to German and English to Chinese translation that human translators vary in performance, with each human translator represented in a distinct cluster. Without taking from the significant achievement of systems that have tied with a human translator, this fact should be taken into account when drawing conclusions about human parity. A tie with a single human translator should not be interpreted as a tie with human performance in general.

4 Test Suites

"Test Suites" have now become an established part of WMT News Translation. Their purpose is to complement the standard one-dimensional manual evaluation. Each test suite can focus on any aspect of translation quality and any subset of language pairs and MT systems.

Anyone can propose their own test suite and take part, and we also try to solicit evaluation from past successful test suite teams to support some cross-year insight.

Each team in the test suites track provides source texts (and optionally references) for any language pair that is being evaluated by WMT News Task. We shuffle these additional texts into the inputs of News Task and ship them as inputs to MT system developers jointly with the regular news texts. The shuffling happens at the document or sentence level as agreed with the test suite authors. (Shuffling at the level of sentences can lead to a very high number of documents in the final test set because each sentence is treated as a separate document.)

MT system developers may decide to skip these documents based on their ID but most of them process test suites along with the main news texts. After collecting the output translations from all WMT News Task Participants, test suites translations are made available back to the test suite authors for evaluation. Test suite sentences do not go through the manual evaluation as described in Section 3.

As in the previous years, test suites are not limited to the news domain, so News Task system may actually underperform on them.

4.1 Test Suite Details

The following paragraphs briefly describe each of the test suites. Please refer to the respective paper for all the details of the evaluation.

4.1.1 Covid Test Suite TICO-19

The TICO-19 test suite was developed to evaluate how well can MT systems handle the newly-emerged topic of COVID-19. Accurate automatic translation can play an important role in facilitating communication in order to protect at-risk populations and combat the *infodemic* of misinformation, as described by the World Health Organization. The test suite has no corresponding paper so its authors provided an analysis of the outcomes directly here.

The submitted systems were evaluated using the test set from the recently-released TICO-19 dataset (Anastasopoulos et al., 2020). The dataset provides manually created translations of COVID-19 related data. The test set consists of PubMed articles (678 sentences from 5 scientific articles), patient-medical professional conversations (104 sentences), as well as related Wikipedia articles (411 sentences), announcements (98 sentences from Wikisource), and news items (67 sentences from Wikinews), for a total of 2100 sentences.

Table 15 outlines the BLEU scores by each submitted system in the English-to-X directions, also breaking down the results per domain. The analysis shows that some systems are significantly more prepared to handle highly narrow-domain data. In

English→Chinese

Ave.	Ave. z	System
80.6	0.568	HUMAN-B
82.5	0.529	HUMAN-A
80.0	0.447	OPPO
79.0	0.420	Tencent-Translation
77.3	0.415	Huawei-TSC
77.4	0.404	NiuTrans
77.7	0.387	SJTU-NICT
76.6	0.373	VolcTrans
73.7	0.282	Online-B
73.0	0.241	Online-A
69.5	0.136	dong-nmt
68.5	0.135	Online-Z
70.1	0.122	Online-G
68.7	0.082	zlabs-nlp

English→Czech

Ave.	Ave. z	System
85.6	0.654	HUMAN
82.2	0.546	CUNI-DocTransformer
81.8	0.538	OPPO
80.8	0.505	SRPOL
80.5	0.458	CUNI-T2T-2018
80.4	0.441	eTranslation
79.3	0.434	CUNI-Transformer
77.1	0.322	UEDIN-CUNI
70.5	0.048	Online-B
69.1	0.017	Online-Z
68.7	0.008	Online-A
62.7	−0.216	Online-G
48.1	−0.760	zlabs-nlp

English→German

Ave.	Ave. z	System
90.5	0.569	HUMAN-B
87.4	0.495	OPPO
88.6	0.468	Tohoku-AIP-NTT
85.7	0.446	HUMAN-A
84.5	0.416	Online-B
84.3	0.385	Tencent-Translation
84.6	0.326	VolcTrans
85.3	0.322	Online-A
82.5	0.312	eTranslation
84.2	0.299	HUMAN-paraphrase
82.2	0.260	AFRL
81.0	0.251	UEDIN
79.3	0.247	PROMT-NMT
77.7	0.126	Online-Z
73.9	−0.120	Online-G
68.1	−0.278	zlabs-nlp
65.5	−0.338	WMTBiomedBaseline

English→Inuktitut (News only)

Ave.	Ave. z	System
90.5	0.574	HUMAN
75.3	0.425	MultiLingual-Ubiqus
77.4	0.409	CUNI-Transfer
71.9	0.369	NRC
74.6	0.368	Facebook-AI
79.2	0.364	NICT-Kyoto
71.6	0.339	Groningen
75.2	0.296	Helsinki
72.8	0.282	SRPOL
68.9	0.084	UQAM-TanLe
66.4	0.081	UEDIN
48.2	−0.384	OPPO

English→Japanese

Ave.	Ave. z	System
79.7	0.576	HUMAN
77.7	0.502	NiuTrans
76.1	0.496	Tohoku-AIP-NTT
75.8	0.496	OPPO
75.9	0.492	ENMT
71.8	0.375	NICT-Kyoto
71.3	0.349	Online-A
70.2	0.335	Online-B
63.9	0.159	zlabs-nlp
59.8	0.032	Online-Z
53.9	−0.132	SJTU-NICT
52.8	−0.164	Online-G

English→Polish

Ave.	Ave. z	System
88.6	0.672	HUMAN
76.4	0.493	SRPOL
75.6	0.435	eTranslation
76.3	0.383	VolcTrans
74.0	0.348	Tilde
70.6	0.316	Online-G
72.0	0.310	OPPO
72.4	0.299	NICT-Kyoto
69.7	0.272	Tilde
71.8	0.255	CUNI-Transformer
70.1	0.236	Online-B
69.0	0.219	SJTU-NICT
64.5	0.097	Online-A
63.9	−0.060	Online-Z
47.7	−0.538	zlabs-nlp

English→Russian

Ave.	Ave. z	System
91.8	0.681	HUMAN
81.5	0.469	Online-G
83.7	0.461	OPPO
79.6	0.404	ariel xv
80.3	0.336	Online-B
75.1	0.252	PROMT-NMT
76.2	0.222	DiDi-NLP
75.3	0.081	Online-A
71.3	0.035	zlabs-nlp
68.5	0.012	Online-Z

English→Tamil

Ave.	Ave. z	System
83.4	0.762	HUMAN
79.0	0.663	Facebook-AI
75.5	0.514	GTCOM
77.3	0.491	Online-B
77.4	0.480	OPPO
78.0	0.457	Online-A
76.7	0.424	VolcTrans
72.8	0.326	Online-Z
72.7	0.307	zlabs-nlp
72.2	0.296	Microsoft-STC-India
74.1	0.231	UEDIN
71.9	0.153	Groningen
68.1	−0.006	DCU
58.2	−0.407	TALP-UPC
53.8	−0.716	Online-G
49.6	−0.819	SJTU-NICT

English→Khmer

Ave.	Ave. z	System
77.4	0.478	GTCOM
76.1	0.435	Online-B
74.6	0.386	Huawei-TSC
73.3	0.349	HUMAN
71.1	0.266	VolcTrans
63.8	0.059	Online-Z
60.9	−0.061	OPPO
57.0	−0.164	Online-Z

English→Pashto

Ave.	Ave. z	System
73.0	0.244	GTCOM
71.9	0.180	Huawei-TSC
70.4	0.162	OPPO
69.7	0.158	Online-B
68.8	0.092	HUMAN
67.7	0.055	Online-Z
66.9	−0.029	VolcTrans

Table 13: Official results of WMT20 News Translation Task for translation out-of-English. Systems ordered by DA score z-score; systems within a cluster are considered tied; lines indicate clusters according to Wilcoxon rank-sum test $p < 0.05$; grayed entry indicates resources that fall outside the constraints provided.

German→ French			French → German		
Ave.	Ave. z	System	Ave.	Ave. z	System
90.4	0.279	OPPO	89.8	0.334	VolcTrans
90.2	0.266	VolcTrans	89.7	0.333	OPPO
89.7	0.262	IIE	89.1	0.319	IIE
89.2	0.243	HUMAN	89.0	0.295	Online-B
89.1	0.226	Online-B	87.4	0.247	HUMAN
89.1	0.223	Online-A	87.3	0.240	Online-A
88.5	0.208	Online-G	87.1	0.221	SJTU-NICT
			86.8	0.195	Online-G
			85.6	0.155	Online-Z

Table 14: Official results of WMT20 News Translation Task for translation from French ↔ German. Systems ordered by DA score z-score; systems within a cluster are considered tied; lines indicate clusters according to Wilcoxon rank-sum test $p < 0.05$; grayed entry indicates resources that fall outside the constraints provided.

addition, the variance of the output quality across languages and across domains highlights the importance of building MT systems that can generalize across domains.

4.1.2 Document Coherence Check via Markable Annotation (Zouhar et al., 2020)

The test suite provided in 2020 by the ELITR project (Zouhar et al., 2020) follows upon Vojtěchová et al. (2019). The focus this year is on "markables", i.e. mainly domain-specific terms that have to be translated consistently and unambiguously throughout the whole document (except news where style may require variation) to maintain lexical coherence. Manual annotation of the translation of markables is contrasted with manual annotation of fluency and adequacy and also BLEU scores.

The test suite is limited to 4 English→Czech documents and 2 Czech→English documents, covering 215 markable occurrences across 4 different domains. The set of markables was collected in the first phase of the annotation, which amounted to 4k assessments across the systems. The second annotation phase with 6.5k assessments compared markable translations, always checking outputs of all the 13 competing MT systems but still considering the document-level context of each of them.

Among other things, the observations indicate that the better the system, the lower the variance in manual scores. Markables annotation then confirms that frequent errors like bad translation of a term need not be the most severe and conversely, even rare errors such as bad disambiguation, over-translation or disappearance of a term or its translation which conflicts with other terms in the document can be critical.

The comparison of MT outputs with the reference (hidden among MT systems) in the evaluation is also interesting. Man-made errors were always marked as less severe than those of MT. The annotation also suggests that one of the document-level systems outperformed the reference in markable evaluation if error severity and frequency are weighted equally.

Fluency and adequacy collected as average sentence-level scores (with access to the full documents of all systems) are curious, revealing perhaps more about the annotators than the MT systems.

4.1.3 Gender Coreference and Bias (Kocmi et al., 2020)

The test suite by Kocmi et al. (2020) focuses on the gender bias in professions (e.g. physician, teacher, secretary) for the translation from English into Czech, German, Polish and Russian. These nouns are ambiguous with respect to gender in English but exhibit gender in the examined target languages.

The test suite is based on the fact that a pronoun referring to the ambiguous noun can reveal the gender of the noun in the English source sentence. Once disambiguated, the gender needs to be preserved in translation. To correctly translate the given noun, the translation system thus has to correctly resolve the coreference link and transfer information from the pronoun to the noun in the

en→◌		Translation Accuracy by Domain (BLEU)				
	Overall	PubMed	Conv.	Wikisource	Wikinews	Wikipedia
Mandarin Chinese (zh)						
SJTU-NICT	57.83	68.88	41.49	33.57	55.97	53.45
OPPO	40.80	49.54	17.01	26.42	31.41	37.53
Online-B	39.55	53.92	23.22	26.09	34.13	32.65
Online-A	35.23	42.81	18.15	20.83	27.77	32.46
Online-G	33.14	38.08	13.06	20.80	26.28	31.74
zlabs-nlp	24.17	31.15	10.11	17.39	21.05	21.00
Online-Z	22.69	28.58	13.31	13.70	17.80	20.30
Khmer (km)						
Online-B	9.01	11.85	7.68	25.86	12.78	6.22
VolcTrans	7.57	12.93	2.35	21.11	4.30	4.63
Online-Z	7.29	9.08	3.38	20.94	5.27	5.65
OPPO	6.99	7.59	6.95	10.73	5.52	6.54
Online-G	2.72	3.10	3.60	1.13	1.70	2.59
Tamil (ta)						
Online-B	30.42	21.42	17.91	31.31	34.11	35.50
Facebook_AI	15.56	12.41	8.71	16.06	16.67	17.40
Online-A	14.49	12.03	7.85	14.78	13.93	16.00
OPPO	12.86	10.22	5.89	13.26	11.67	14.51
UEDIN	12.25	10.15	9.59	12.90	13.83	13.36
Microsoft_STC_India	11.91	9.48	6.49	12.07	12.56	13.33
Online-Z	11.70	9.45	10.87	13.52	10.10	12.96
VolcTrans	11.63	10.12	11.91	9.52	12.32	12.53
zlabs-nlp	10.32	8.91	5.85	9.64	10.90	11.20
DCU	9.70	7.66	7.79	8.44	9.36	10.91
Groningen	8.93	8.00	5.95	8.14	9.66	9.47
Online-G	7.32	6.79	8.42	8.32	5.59	7.58
TALP_UPC	6.25	5.77	3.48	5.47	7.32	6.54
SJTU-NICT	2.91	3.01	3.72	5.26	2.68	2.67
Pashto (ps)						
Online-B	36.56	49.26	26.94	12.15	8.85	32.25
VolcTrans	18.47	24.22	16.21	12.58	8.96	16.41
OPPO	18.24	21.88	13.98	14.40	7.98	17.15
Online-Z	15.14	18.59	13.57	12.87	7.60	13.93
Russian (ru)						
Online-B	40.20	29.71	26.37	22.90	40.44	46.38
Online-G	33.78	28.20	25.51	22.58	32.39	37.30
PROMT_NMT	32.69	27.45	24.82	21.90	30.39	36.05
ariel197197	32.40	25.44	28.33	22.17	37.04	35.96
OPPO	31.86	29.04	23.33	22.17	32.27	33.76
Online-A	29.84	24.76	21.13	20.53	27.54	33.07
zlabs-nlp	25.83	23.63	21.96	19.40	25.97	27.20
Online-Z	24.67	20.26	20.43	20.01	26.09	27.07

Table 15: TICO-19 test suite results on the English-to-X WMT20 translation directions.

antecedent (a less common direction of information flow), and then correctly express the noun in the target language. The success of the MT system in this test can be established automatically, whenever the gender of the target word can be automatically identified.

Kocmi et al. (2020) build upon the WinoMT (Stanovsky et al., 2019) test set, which provides exactly the necessary type of sentences containing an ambiguous profession noun and a personal pronoun which unambiguously (for the human eye) refers to it based the situation described. When extending WinMT with Czech and Polish, Stanovsky et al. have to disregard some test patterns but the principle remains.

The results indicate that *all* MT systems fail in this test, following gender bias (stereotypical patterns attributing the masculine gender to some professions and feminine gender to others) rather than the coreference link.

4.1.4 Linguistic Evaluation of German-to-English (Avramidis et al., 2020)

The test suite by DFKI covers 107 grammatical phenomena organized into 14 categories. Since 2018, the same set of phenomena are being tested annually (Macketanz et al., 2018; Avramidis et al., 2019).

Automatic evaluation is complemented with 45 hours of human annotation.

This year, the newcomers VOLCTRANS and TOHOKU-AIP-NTT perform particularly well in the tested phenomena, followed by the traditional systems UEDIN, ONLINE-B, ONLINE-G, and ONLINE-A.

The generally good news is that systems which participated in both WMT19 and WMT20 show an improvement this year. Given that the test suite target side remains undisclosed, these scores can be deemed absolute, unlike the official DA scores which are only relative within each year and set of systems.

The test suite allows to report these improvements per linguistic category and specifically for each MT system that participated in two consecutive years. The biggest improvements are observed in long distance dependencies or interrogatives, verb valency, ambiguity and punctuation, and we tend to attribute all these improvements to increased capacity (which allows increased sensitivity to long-range relations) of the models.

4.1.5 Word Sense Disambiguation (Scherrer et al., 2020b)

Scherrer et al. (2020b) is a followup of last year's evaluation (Raganato et al., 2019), assessing the ability of MT systems to disambiguate a word given its context of the sentence.

The underlying MuCoW (multilingual contrastive word sense disambiguation) dataset contains approximately 2k to 4k sentences per language pair selected from large parallel corpora to contain particularly ambiguous words.

This year, the focus was on language pairs that appeared both in WMT19 and WMT20 (and were available in the MuCoW dataset), namely English→Czech, English↔German, and English→Russian.

Comparing overall numbers across the years, Scherrer et al. (2020b) report that ambiguous words are correctly disambiguated in the majority of cases. Both precision (percentage of correct choices out of sentences where either good or bad expected translation was found) and recall (percentage of correct choices out of all sentences) are above 60 % and reaching 80 % for the best systems in a given language pair when mixing "in-domain" and "out-of-domain" evaluation. The "out-of-domain" synsets are those that are represented in the test suite with more than half of cases coming from the colloquial subtitle domain; other synsets are deemed "in-domain". The "in-domain" scores are generally higher, with precisions above 95 % for the best Czech and Russian systems. Across the years, no real improvement is however observed.

Three cases suggest that training systems at the level of documents decreases their performance in this sentence-level evaluation (each sentence forms a separate document): DocTransformer vs. Transformer by CUNI in 2019 and 2020 and Microsoft document-level vs. sentence-level submission in 2019.

5 Similar Language Translation

Most shared tasks at WMT (e.g. News, Biomedical) have historically dealt with translating texts from and to English. In recent years, we observed a growing interest in training systems to translate between languages other than English. This includes a number of papers applying MT to translate between pairs of closely-related languages, national language varieties, and dialects

of the same language (Zhang, 1998; Marujo et al., 2011; Hassani, 2017; Costa-jussà et al., 2018; Popović et al., 2020). To address this topic, the first Similar Language Translation (SLT) shared task at WMT 2019 has been organized. It featured data from three pairs of closely-related languages from different language families: Spanish - Portuguese (Romance languages), Czech - Polish (Slavic languages), and Hindi - Nepali (Indo-Aryan languages).

Following the success of the first SLT shared task at WMT 2019 and the interest of the community in this topic, we organize, for the second time at WMT, this shared task to evaluate the performance of state-of-the-art translation systems on translating between pairs of languages from the same language family. SLT 2020 features five pairs of similar languages from three different language families: Indo-Aryan, Romance, and South-Slavic. Translations were evaluated in both directions using automatic evaluation metrics presented in this section.

5.1 Data

Training We have made available a number of data sources for the SLT shared task. Some training datasets were used in the previous editions of the WMT News Translation shared task and were updated (Europarl v10, News Commentary v15, Wiki Titles v2), while some corpora were newly introduced (JRC Acquis). The released parallel HI–MR dataset was collected from news (Siripragada et al., 2020), PMIndia (Haddow and Kirefu, 2020) and Indic Wordnet (Bhattacharyya, 2010; Kunchukuttan, 2020a) datasets. All data were initially combined, tokenized using indic-nlp tokenizer (Kunchukuttan, 2020b) and randomly shuffled. From the combined corpus, we randomly extracted 49,434 sentences for the training set and the rest are used as development and test sets. For the South-Slavic language pairs we used large datasets available from Opus (Tiedemann and Nygaard, 2004)[11], more precisely the OpenSubtitles, MultiParaCrawl, DGT and JW300 data. Different to the other language groups, for monolingual data web corpora of the three languages (Ljubešić and Erjavec, 2011; Ljubešić and Klubička, 2014; Erjavec et al., 2015) were given to the participants.

Development and Test Data The development and test sets for Spanish–Catalan and Spanish–Portuguese language pairs were created from a corpus provided by Pangeanic[12]. First, we performed cleaning using CLEAN-CORPUS-N.PERL[13] script to retain sentences that have between 4 and 100 tokens. This narrowed the number of sentences to 1,287 and 1,535 in dev and test sets respectively. Finally, sentences containing metadata information were removed, which resulted in 1,283 and 1,495 sentences in dev and test sets respectively.

The aforementioned shuffled combined HI–MR dataset, 1411 sentences are used for development set and 3882 for the test set. Finally, the test set was equally split into two different test sets: 1941 sentences used for HI to MR and 1941 sentences were used for MR to HI.

For the Slovene–Croatian and Slovene–Serbian language pairs, development and test data were obtained from the Ciklopea translation agency[14] in form of a data donation from the Bisnode business intelligence company[15]. The data consists of public relations releases translated in various directions between the three languages. The data was cleaned, deduplicated and shuffled, resulting in 2,457 dev and 2,582 instances for the Slovene–Croatian pair, and 1,259 dev and 1,260 test instances in the Slovene–Serbian pair. Given that these translations sometimes form Slovene–Croatian–Serbian triangles, special care was invested in circumventing data leakage between development data on one side, and test data on the other, of the two language pairs.

5.2 Participants and Approaches

The second edition of the WMT SLT task attracted 68 teams who signed up to participate in the competition and 18 of them submitted their system outputs. In the end of the competition, 14 teams submitted system description papers which are referred to in this report. Table 22 summarizes the participation across language pairs and translation directions and includes references to the 14 system description papers.

Next we provide summaries for each of the entries we received:

A3-108 The team A3-108 submitted their system for HI–MR and MR–HI. The team initially

[11] http://opus.nlpl.eu/

[12] https://www.pangeanic.com/

[13] https://github.com/moses-smt

[14] https://ciklopea.com

[15] https://www.bisnode.hr

Table 16: Corpora for the Hindi ↔ Marathi language pair.

		Corpus	Sentences
Parallel	Hindi ↔ Marathi	News	12,349
	Hindi ↔ Marathi	PM India	25,897
	Hindi ↔ Marathi	Indic WordNet	11,188
Monolingual	Hindi	News Crawl 2008-2019	32,609,161
	Hindi	IITB	45,075,242
	Hindi	hi.yyyy_nn.raw.xz 2012-2017	
	Marathi	News Crawl 2018-2019	326,748
	Marathi	mr.yyyy_nn.raw.xz 2012-2017	
Dev	Hindi ↔ Marathi		1,411
Test	Hindi ↔ Marathi		1,941

Table 17: Corpora for the Spanish ↔ Catalan language pair.

		Corpus	Sentences
Parallel	Spanish ↔ Catalan	Wiki Titles v2	446,326
	Spanish ↔ Catalan	DOGC v2	10,933,622
Monolingual	Spanish	Europarl v10	2,038,042
	Spanish	News Commentary v15	465,165
	Spanish	News Crawl 2007-2019	53,874,815
	Catalan	caWaC	24,745,986
Dev	Spanish ↔ Catalan		1,283
Test	Spanish ↔ Catalan		1,495

Table 18: Corpora for the Spanish ↔ Portuguese language pair.

		Corpus	Sentences
Parallel	Spanish ↔ Portuguese	Europarl v10	1,801,845
	Spanish ↔ Portuguese	News Commentary v15	48,259
	Spanish ↔ Portuguese	Wiki Titles v2	649,833
	Spanish ↔ Portuguese	JRC-Acquis	1,650,126
Monolingual	Spanish	Europarl v10	2,038,042
	Spanish	News Commentary v15	465,165
	Spanish	News Crawl 2007-2019	53,874,815
	Portuguese	Europarl v10	2,016,635
	Portuguese	News Commentary v15	73,550
	Portuguese	News Crawl 2008-2019	9,392,574
Dev	Spanish ↔ Portuguese		1,283
Test	Spanish ↔ Portuguese		1,495

Table 19: Corpora for the Slovenian ↔ Croatian language pair.

		Corpus	Sentences
Parallel	Slovenian ↔ Croatian	OpenSubtitles v2018	15,636,933
	Slovenian ↔ Croatian	MultiParaCrawl v5	271,415
	Slovenian ↔ Croatian	JW300 v1	1,052,547
	Slovenian ↔ Croatian	DGT v2019	698,314
Monolingual	Slovenian	slWaC	46,251,729
	Croatian	hrWaC	64,577,734
Dev	Slovenian ↔ Croatian		2,457
Test	Slovenian ↔ Croatian		2,582

Table 20: Corpora for the Slovenian ↔ Serbian language pair.

		Corpus	Sentences
Parallel	Slovenian ↔ Serbian	OpenSubtitles v2018	16,426,054
Monolingual	Slovenian	slWaC	46,251,729
	Serbian	srWaC	24,073,253
Dev	Slovenian ↔ Serbian		1,259
Test	Slovenian ↔ Serbian		1,260

Table 21: Corpora for the Croatian ↔ Serbian language pair.

		Corpus	Sentences
Parallel	Croatian ↔ Serbian	SETimes	203,989

build SMT models for both language direction after three steps preprocessing: (i) default – indic_nlp_library[16] and moses tokenizer[17], (ii) morfessor[18] and (iii) BPE[19]. These SMT models were used for back-translation. Finally, these back-translation data were used to train their NMT system.

ADAPT-DCU The ADAPT-DCU team participated in the SLT task on the Croatian–Slovene and Serbian–Slovene language pairs. The team's submissions were based on the Sockeye implementations of the Transformer, with a joint 32k-large BPE vocabulary for all three languages. The submission were regularly multilingual (having Slovene on one side and Croatian and Serbian on the other). The team used only OpenSubtitles bilingual training data, considering other available data to be too noisy. The basic implementation of the multilingual system was submitted as the second contrastive system, the multilingual implementation trained on filtered parallel data as the first contrastive system, while the primary submission included backtranslation of target monolingual data of segments similar to the development data. By performing n-gram-character-based filtering of training data the training time was cut in half with a minor improvement on the translation quality, while the largest improvements in translation quality were obtained by back-translating data similar to development data (between 8 and 14 BLEU points).

f1plusf6 During preprocessing as Marathi and Hindi are rich in terms of morphology, Applied two way segmentation as preprocessing, first supervised and unsupervised word based morphological segmentation and then BPE based segmentation to tackle low-resource language pairs. The participants used shared vocab across training and utilised POS based features on the source side to create initial models for both directions.

For preparing unsupervised back-translation parallel data they used aligned embedding space to generate word by word parallel sentences for both language directions. They also prepared initial models from the provided parallel data for back translation from monolingual data and pruned back-translation pairs based on perplexity score. Their model is based on Luong's attention on bi-LSTM network, copy attention on dynamically generated dictionary with label smoothing and dropouts to reduce overfitting.

Fast-MT Fast-MT team submitted their NMT system where Transformers and Recurrent Attention models are effectively used. They combined the recurrence based layered encoder-decoder model with the Transformer model. Their submitted system for Indo-Aryan Language (Hindi to Marathi) pair is trained on the parallel corpus of the training dataset provided by the organizers.

IIAI IIAI TEAM participate in both directions of the Hindi–Marathi translation task. Their primary submission is a transformer model trained on the released parallel and back-translated monolingual data. The team jointly learned BPE from the merged source–target corpus. After BPE, sentences were corrupted and reconstructed using the two ways:(i) 15% of the subwords in the sentence are randomly selected and masked, (ii) 15% of the subwords are randomly selected one by one and swapped with another randomly-selected subword in the sequence.

[16] https://anoopkunchukuttan.github.io/indic_nlp_library/
[17] https://github.com/moses-smt/mosesdecoder
[18] https://github.com/aalto-speech/morfessor
[19] https://github.com/rsennrich/subword-nmt

Team	System Description Paper
A3108	Yadav and Shrivastava (2020)
ADAPT-DCU	Popović and Poncelas (2020)
f1plusf6	Mujadia and Sharmaa (2020)
FAST-MT	Dhanani and Rafi (2020)
IIAI	
IIT-DELHI	Madaan et al. (2020)
INFOSYS	Rathinasamy et al. (2020)
IPN-CIC	Menéndez-Salazar et al. (2020)
NICT	
NITS-CNLP	Laskar et al. (2020)
NLPRL	Kumar et al. (2020)
NLPRL-IITBHU	
NUIG-Panlingua-KMI	Ojha et al. (2020)
NUST_FJWU	Haq et al. (2020)
Prompsit	
UBC-NLP	Adebara et al. (2020)
UPCTALP	Boncompte and Costa-jussà (2020)
WIPRO-RIT	Pal and Zampieri (2020)

Table 22: The teams that participated in the SLT 2020 task and their system description papers.

IITDELHI Team IITDELHI participated in the SLT task on Hindi–Marathi and Spanish–Portuguese language pairs. The team's primary submission builds on fine-tuning over pretrained mBART. They used pre-trained weights of the mBART model (Liu et al., 2019), which is pre-trained on large amounts of monolingual data for 25 languages including Spanish, however Portuguese is not there. The authors initialized a Transformer architecture with 12 encoder and decoder layers using the pre-trained weights, and then directly fine-tuned with the released training data. The authors conclude that mBART is helpful for transfer learning, even though the languages that are not available in the pre-trained model.

INFOSYS Infosys system for Hindi–Marathi (Primary) task is designed to learn the nuances of translation of this low resource language pair by taking advantage of the fact that the source (Hindi) and target (Marathi) languages are same alphabet languages. This system is an ensemble of FairSeq model built on anonymized parallel data and FairSeq back-translation model. The common words/tokens between source and target languages are anonymized during pre-processing upon which the FairSeq model is trained. The input statements during inferencing are anonymized based on the vocabulary of common tokens prepared during training and the predicted statements are de-anonymized during post-processing accordingly. This improved the accuracy (BLEU) of FairSeq considerably. Pre-processing also applies traditional parallel corpus filtering techniques to clean parallel data followed by domain specific techniques. There were records containing multiple statements delimited by slashes, where the domain specific techniques are applied to transform them in to records that retain only the matching single statement, identified based on its syntactic similarity with its parallel statement. Synthetic data generated with the mono-lingual (Marathi) data during FairSeq back-translation has unknown words (w.r.t parallel data vocabulary), resulting unknown words during prediction, which are downvoted while ensembling.

IPN-CIC This team participated in the Spanish-Portuguese language pair. The systems used the Transformer architecture with a fine-tuning for domain adaptation. The team proposed experiments on the kind of tokens used (words and sub-word units) and the initialization of the word embeddings in the systems using either a random initialization or pre-trained word embeddings.

NICT NICT participated in two language pairs: Hindi–Marathi and Spanish–Catalan, for both translation directions. Their primary submission is an unsupervised NMT system, initialized with a pre-trained cross-lingual language model (XLM), that has been trained using only the monolingual data provided by the organizers. They used the

standard hyper-parameters for training XLM and unsupervised NMT. Their contrastive submission is the same but supervised NMT system trained on the combination of the released bilingual and monolingual data.

NITS-CNLP NITS-CNLP system for HI–MR and MR–HI translation is based on cross-lingual language modelling with masked language modeling and translation language modeling. These language models were pre-trained on monolingual corpus and fine-tuned on parallel data following the architecture of Conneau and Lample (2019) and employing 6 layers with 8 attention heads and with 32 batch size, trained on a single GPU.

NLPRL This system submitted by the NLPRL team for the HI-MR is based on the Transformer approach. The system were trained on only the released parallel corpus. The team used SentencePiece library for preprocessing and set vocabulary size of 5000 symbols for source and target byte-pair encoding, respectively.

NLPRL-BHU The team participated in the HI $\leftrightarrow$ MR language pair. The participants used byte pair encoding to preprocess the data and fairseq library with the GRU-transformer for training.

NUIG-Panlingua-KMI The NUIG-Panlingua-KMI team explored phrase-based SMT, dependecy-based SMT method and neural method (used subword) for Hindi$\leftrightarrow$Marathi language pair.

NUST-FJWU NUST-FJWU system is an extension of state-of-the-art Transformer model with hierarchical attention networks to incorporate contextual information. During training the model used back-translation.

Prompsit This team is participating with a rule-based system based on Apertium (Forcada et al., 2009-11). Apertium is a free/open-source platform for developing rule-based machine translation systems and language technology that was first released in 2005. Apertium is hosted in Github where both language data and code are licensed under the GNU GPL. It is a research and business platform with a very active community that loves small languages. Language pairs are at a very different level of development and output quality in the platform, depending on two main variables: how much funded or in-kind effort has been devoted to it and the nature of the languages itself (the closer, the better).

UBC-NLP The UBC-NLP team participated in the SLT task on all the available language pairs. The team regularly used all the parallel data and trained 6-layer Transformer models based on the Fairseq library. Only for the Slovene–Croatian language pair the team performed back-translation, noticing a 3 BLEU point improvement in the results. This team obtained better results with bilingual than with multilingual models (training a single model for all language groups).

UPCTALP The UPCTALP participated in the Romance pairs. This team made use of the Transformer architecture improved with multilingual, back-translation and fine-tuning techniques. Each of this techniques improved over the previous one.

WIPRO-RIT WIPRO-RIT submitted their system to the SLT 2020 Indo-Aryan track. The presented system is a single multilingual NMT system based on the transformer architecture that can translate between multiple languages. The presented model is inspired from the model described in Johnson et al. (2017). WIPRO-RIT achieved competitive performance ranking 1st in Marathi to Hindi and 2nd in Hindi to Marathi translation among 22 systems.in Hindi to Marathi translation among 22 systems.

5.3 Results

We present results for the three language families: three different language families: Indo-Aryan (Hindi - Marathi), Romance (Spanish - Catalan, Spanish - Portuguese), and South-Slavic (Slovene - Croatian, Slovene - Serbian), all of them in the two possible directions. Like last year edision, the second edition of the Similar Translation Task evaluation was also performed on automatic basis using BLEU (Papineni et al., 2002), RIBES (Isozaki et al., 2010) and TER (Snover et al., 2006) measures. Each language direction is reported in one different table which contain information of the team; type of system, either contrastive (CONTRASTIVE) or primary (PRIMARY), and the BLEU, RIBES and TER results. The scores are sorted by BLEU. In general, primary systems tend to be better than contrastive systems, as expected, but there are some exceptions.

This year we recived major number of participants for the case of Indo-Aryan language group

i.e. Hindi–Marathi (in both directions). We received 22 submissions from 14 teams. The best systems (INFOSYS) based on BLEU for Hindi–Marathi achieved score 18.26, however based on other evaluation matric WIPRO-RIT achieved the best 62.45 RIBES and around 72 TER (see Table 23). While in the other direction Marathi–Hindi the best performing system (WIPRO-RIT) reached 24.53 of BLEU and 66.39 on TER, but based on RIBES score 66.83, IITDELHI performed the best (see Table 24).

Similarly to the previous edition of the SLT shared task, participants could submit systems for the Spanish–Portuguese language pair (in both directions). The best systems for Spanish-to-Portuguese achieved over 32 BLEU and around 52 TER. While in the opposite direction (Portuguese-to-Spanish) the best performing system reached 33.82 of BLEU, but its TER score was 52.41, which is higher than in the case of best performing Spanish-to-Portuguese systems. As the Spanish–Catalan dev and test sets were aligned with Spanish–Portuguese ones, we noticed that the best results for the Spanish–Catalan language pair are in general much better than for Spanish–Portuguese. For Spanish-to-Catalan the best system attained over 86 BLEU and below 8 TER. In the case of Catalan-to-Spanish, the best systems scored around 77 BLEU and less than 15 TER.

A new language group in this year's SLT task is the group of (Western) South Slavic languages - Slovene, Croatian and Serbian, forming two language pairs - Slovene–Croatian and Slovene–Serbian, with one additional twist given the very high mutual intelligibility of Croatian and Serbian. The best systems for Slovene-to-Croatian achieved 36 BLEU and 43 TER, which is significantly worse than the results of the same best-performing system in the opposite direction - 43 BLEU and 36 TER. On the Slovene–Serbian pair a similar phenomenon can be observed - Slovene to Serbian achieving 39 BLEU and 40 TER, while the opposite direction achieves 47 BLEU and 33 TER. The reason for such a significant lack of symmetry is the better performance of the systems translating into Slovene, probably given that (Croatian and Serbian) multi-source translation (into Slovene) is simpler than multi-target translation, which was, finally, propagated to the back-translation procedure, increasing the difference between the directions even further.

5.4 Summary

In this section, we presented the results of the WMT SLT 2020 task. The second iteration of this competition featured data from five language pairs from three different language families: Hindi-Marathi; Spanish-Catalan and Spanish-Portuguese; Sloven-Croatian and Slovene-Serbian. We evaluated the systems translating in both directions of the language pair using three automatic metrics: BLEU, RIBES, and TER. We observed that the performance varies widely between language pairs. For example. the best performing systems trained to translate between Catalan and Spanish in both directions obtained significantly higher results than those trained to translate between other language pairs.

In terms of participation, SLT received system submissions from 18 teams. In the end of the competition, 14 teams wrote system description papers that appear in the WMT proceedings. The list of teams with references to the respective system description paper is presented in Table 22. Finally, short summaries of each entry, based on the description provided by the participants, were also presented in this section.

6 Conclusion

This paper presented the results of WMT20 news translation and similar language translation shared tasks, as well as the extra test suites added to the news translation task. Our main findings rank participating systems in their sentence-level and document-level translation quality, as assessed in a large-scale manual evaluation using the method of Direct Assessment (DA).

For out-of-English language pairs, DA was modified so that the context of the whole document is available while judging individual sentences and assessors are allowed to return to any sentence judgement within the document.

As in previous years, the effect of translationese (translating from a source which itself was produced in translation) was avoided except lower-resourced Inuktitut↔English, Pashto↔English, Khmer↔English, and German↔French by creating reference translations always in the same direction as the MT systems are run. Furthermore, 8 out-of-English language pairs would not need human reference for our evaluation at all because the assessors are evaluating translation candidates bilingually, comparing them to the source

Team	Type	BLEU ↑	RIBES ↑	TER ↓
INFOSYS	PRIMARY	18.26	56.73	76.48
WIPRO-RIT	PRIMARY	16.62	62.45	72.23
WIPRO-RIT	CONTRASTIVE2	15.42	61.02	73.59
IITDELHI	PRIMARY	15.14	61.06	74.63
IIAI	CONTRASTIVE	14.99	52.11	85.77
IITDELHI	CONTRASTIVE	14.91	57.63	81.19
IIAI	PRIMARY	14.73	52.80	86.13
WIPRO-RIT	CONTRASTIVE1	13.25	58.51	76.17
NLPRL	PRIMARY	12.50	58.66	76.86
NITS-CNLP	PRIMARY	11.59	57.76	79.07
A3108	PRIMARY	11.41	57.20	79.96
A3108	CONTRASTIVE	10.21	55.17	82.01
NUIG-Panlingua-KMI	CONTRASTIVE	9.76	52.18	91.49
NUIG-Panlingua-KMI	PRIMARY	9.38	51.88	91.24
f1plusf6	PRIMARY	5.49	43.74	94.60
f1plusf6	CONTRASTIVE	5.41	43.49	94.52
FAST-MT	PRIMARY	3.68	31.14	97.64
NICT	CONTRASTIVE	3.41	42.43	-
NICT	PRIMARY	1.26	31.20	-
UBC-NLP	PRIMARY	0	1	-
UBC-NLP	CONTRASTIVE	0	0.12	-

Table 23: Results for Hindi to Marathi translation.

Team	Type	BLEU ↑	RIBES ↑	TER ↓
WIPRO-RIT	PRIMARY	24.53	66.23	66.39
IITDELHI	PRIMARY	24.53	66.83	67.25
WIPRO-RIT	CONTRASTIVE2	22.93	65.89	68.11
WIPRO-RIT	CONTRASTIVE1	22.69	65.01	68.13
A3108	CONTRASTIVE	21.11	60.76	77.28
NLPRL	PRIMARY	20.72	64.46	71.04
IIAI	CONTRASTIVE	20.32	59.56	79.32
IIAI	PRIMARY	20.04	58.95	80.27
IITDELHI	CONTRASTIVE	18.74	58.56	77.22
A3108	PRIMARY	18.32	59.31	77.35
f1plusf6	PRIMARY	18.14	60.86	78.27
NUIG-Panlingua-KMI	CONTRASTIVE	17.39	58.84	81.15
NUIG-Panlingua-KMI	PRIMARY	17.38	59.31	81.47
f1plusf6	CONTRASTIVE	17.17	60.69	78.18
NITS-CNLP	PRIMARY	15.44	61.13	75.96
NICT	CONTRASTIVE	11.20	56.13	-
FAST-MT	PRIMARY	9.02	46.96	88.68
NUST_FJWU	CONTRASTIVE	6.79	46.27	91.28
NUST_FJWU	PRIMARY	6.71	43.19	93.74
NICT	PRIMARY	6.28	50.14	-
NLPRL-IITBHU	PRIMARY	0.12	7.66	-
UBC-NLP	PRIMARY	0.09	7.19	-
UBC-NLP	CONTRASTIVE	0	0.09	-

Table 24: Results for Marathi to Hindi translation.

Team	Type	BLEU ↑	RIBES ↑	TER ↓
Prompsit	PRIMARY	77.08	95.71	12.35
NICT	CONTRASTIVE	76.67	93.33	14.22
UPCTALP	PRIMARY	68.84	89.83	20.09
NICT	PRIMARY	68.43	92.13	19.47
UBC-NLP	PRIMARY	0.17	4.81	-
UBC-NLP	CONTRASTIVE	0	1.50	-

Table 25: Results for Catalan to Spanish translation.

Team	Type	BLEU ↑	RIBES ↑	TER ↓
Prompsit	PRIMARY	86.48	97.37	7.716
Prompsit	CONTRASTIVE	81.36	96.64	10.15
UPCTALP	PRIMARY	60.50	90.25	25.80
NICT	CONTRASTIVE	59.05	90.73	25.90
NICT	PRIMARY	51.97	88.30	31.68
UBC-NLP	CONTRASTIVE	9.53	64.17	77.42
UBC-NLP	PRIMARY	8.49	58.93	84.16

Table 26: Results for Spanish to Catalan translation.

Team	Type	BLEU ↑	RIBES ↑	TER ↓
UPCTALP	PRIMARY	33.82	76.04	52.41
IITDELHI	PRIMARY	32.84	74.84	52.65
Prompsit	PRIMARY	30.27	75.37	54.46
IPN-CIC	PRIMARY	28.38	72.24	56.27
IPN-CIC	CONTRASTIVE1	27.98	72.11	56.16
IPN-CIC	CONTRASTIVE2	27.41	75.18	57.28
UBC-NLP	CONTRASTIVE	0.06	1.50	-
UBC-NLP	PRIMARY	0	5.86	-

Table 27: Results for Portuguese to Spanish translation.

Team	Type	BLEU ↑	RIBES ↑	TER ↓
IIT-DELHI	PRIMARY	32.69	74.05	51.74
UPCTALP	PRIMARY	32.33	73.04	52.06
IPN-CIC	PRIMARY	27.08	72.98	55.34
Prompsit	PRIMARY	26.91	75.79	54.63
Prompsit	CONTRASTIVE	26.81	75.71	54.73
IPN-CIC	CONTRASTIVE1	23.91	71.55	57.55
IPN-CIC	CONTRASTIVE2	23.90	73.73	58.07
UBC-NLP	PRIMARY	17.06	52.55	76.21
UBC-NLP	CONTRASTIVE	4.47	52.72	88.13

Table 28: Results for Spanish to Portuguese translation.

Team	Type	BLEU ↑	RIBES ↑	TER ↓
ADAPT-DCU	PRIMARY	43.41	73.77	35.8
ADAPT-DCU	CONTRASTIVE2	29.04	68.71	48.74
ADAPT-DCU	CONTRASTIVE1	26.96	64.02	50.73
UBC-NLP	PRIMARY	0.07	1.03	-
UBC-NLP	CONTRASTIVE	0	0.25	-

Table 29: Results for Croatian to Slovene translation.

Team	Type	BLEU ↑	RIBES ↑	TER ↓
ADAPT-DCU	PRIMARY	35.56	72.04	43.19
ADAPT-DCU	CONTRASTIVE1	27.63	70.53	49.91
ADAPT-DCU	CONTRASTIVE2	23.3	60.8	52.79
UBC-NLP	PRIMARY	22.26	64.41	64.5
UBC-NLP	CONTRASTIVE	1.68	35.35	-

Table 30: Results for Slovene to Croatian translation.

Team	Type	BLEU ↑	RIBES ↑	TER ↓
ADAPT-DCU	PRIMARY	47.45	75.11	32.61
ADAPT-DCU	CONTRASTIVE1	33.5	70.86	44.58
ADAPT-DCU	CONTRASTIVE2	30.28	65.92	47.77
UBC-NLP	CONTRASTIVE	0	0.39	-
UBC-NLP	PRIMARY	0	1.3	-

Table 31: Results for Serbian to Slovene translation.

Team	Type	BLEU ↑	RIBES ↑	TER ↓
ADAPT-DCU	PRIMARY	39.16	73.37	39.81
ADAPT-DCU	CONTRASTIVE1	29.79	70.24	47.55
ADAPT-DCU	CONTRASTIVE2	25.7	64.81	50.51
UBC-NLP	PRIMARY	20.18	63.37	65.56
UBC-NLP	CONTRASTIVE	2.01	38.87	-

Table 32: Results for Slovene to Serbian translation.

text (as opposed to the reference) in these language pairs. The reference translations are nevertheless included as evaluation, hidden among participating MT systems.

This year, English→German included two independent reference translations and one human-produced paraphrase, and English→Chinese included two references. Each of these translations ended up significantly differing in quality from the other ones. In German↔English and also Chinese→English and English→Inuktitut, some MT systems fall in the same cluster with human translation. The observed variance of human translation quality however demands modesty before making any claims about human parity.

The need for cautious interpretation of the results is also strengthened by the fact that even in English→German and English→Czech where human translation was seemingly significantly surpassed in 2018 and/or 2019, the result is not confirmed this year. Furthermore and similarly to previous year, a test suite this year again suggests that some aspects of translation are not handled by current systems at all. This year all MT systems fall into the gender bias trap (Kocmi et al., 2020) and they tend to make more severe errors than humans (Zouhar et al., 2020).

The results of the task on similar language translation indicate that the performance when translating between pairs of closely-related languages is extremely varied across different language pairs. The best performing systems trained to translate between Catalan and Spanish, for example, obtained significantly higher results in both directions than those trained to translate between other language pairs in terms of BLEU, RIBES, and TER.

Acknowledgments

Translation of the test sets for the News task was sponsored by the EU H2020 projects ELITR and Bergamot (English-Czech), and GoURMET (English-Tamil), by Yandex (Russian-English), Microsoft (Chinese-English and German-English), Tilde (Polish-English), Lingua Custodia (French-German), Facebook (Pashto-English, Khmer-English), the University of Tokyo and NTT (Japanese-English). The human evaluation was co-funded by Google and Microsoft. For the human evaluation of English→Inuktitut, we are grateful for the funding received from the NRC and to the workers of the Pirurvik Centre who did the evaluation. We are also grateful to the many workers who contributed to the

human evaluation via Mechanical Turk. We would like to thank Roland Kuhn for advising on the Inuktitut↔English task organization and the Nunavut Maligaliurvia (Legislative Assembly of Nunavut) and Nunatsiaq News for supplying all the Inuktitut↔English parallel data.

The organizers of the similar languages task would like to thank Ciklopea and Bisnode for the Croatian, Serbian, and Slovene data, and Pangeanic for the Catalan, Portuguese, and Spanish data. The work of the organizers of the similar languages task is supported in part by the Spanish Ministerio de Ciencia e Innovación, through the postdoctoral senior grant Ramón y Cajal and by the Agencia Estatal de Investigación through the projects EUR2019-103819, PCIN-2017-079 and PID2019-107579RB-I00 / AEI / 10.13039/501100011033.

This work was supported in part by Science Foundation Ireland in the ADAPT Centre for Digital Content Technology (www.adaptcentre.ie) at Trinity College Dublin funded under the SFI Research Centres Programme (Grant 13/RC/2106) co-funded under the European Regional Development Fund.

Ondřej Bojar would like to acknowledge the grant no. 19-26934X (NEUREM3) of the Czech Science Foundation for his time as well as co-funding manual annotation.

References

Ife Adebara, El Moatez Billah Nagoudi, and Muhammad Abdul Mageed. 2020. Translating similar languages: Role of mutual intelligibility in multilingual transformers. In *Proceedings of WMT*.

Antonios Anastasopoulos, Alessandro Cattelan, Zi-Yi Dou, Marcello Federico, Christian Federman, Dmitriy Genzel, Francisco Guzmán, Junjie Hu, Macduff Hughes, Philipp Koehn, Rosie Lazar, Will Lewis, Graham Neubig, Mengmeng Niu, Alp Öktem, Eric Paquin, Grace Tang, and Sylwia Tur. 2020. Tico-19: the translation initiative for covid-19. In *NLP COVID-19 Workshop*, Online.

Eleftherios Avramidis, Vivien Macketanz, Ursula Strohriegel, Aljoscha Burchardt, and Sebastian Möller. 2020. Fine-grained linguistic evaluation for state-of-the-art machine translation. In *Proceedings of the Fifth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Eleftherios Avramidis, Vivien Macketanz, Ursula Strohriegel, and Hans Uszkoreit. 2019. Linguistic Evaluation of German-English Machine Translation Using a Test Suite. In *Proceedings of the Fourth Conference on Machine Translation*, Florence, Italy. Association for Computational Linguistics.

George Awad, Asad Butt, Keith Curtis, Yooyoung Lee, Jonathan Fiscus, Afzal Godil, Andrew Delgado, Jesse Zhang, Eliot Godard, Luca Diduch, Alan F. Smeaton, Yvette Graham, and Wessel Kraaij. 2019. Trecvid 2019: An evaluation campaign to benchmark video activity detection, video captioning and matching, and video search & retrieval. In *Proceedings of TRECVID*, volume 2019.

Loïc Barrault, Magdalena Biesialska, Marta R. Costa-jussà, Fethi Bougares, and Olivier Galibert. 2020. Findings of the first shared task on lifelong learning machine translation. In *Proceedings of the Fifth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Loïc Barrault, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. 2019. Findings of the 2019 Conference on Machine Translation (WMT19). In *Proceedings of the Fourth Conference on Machine Translation*, Florence, Italy. Association for Computational Linguistics.

Rachel Bawden, Alexandra Birch, Radina Dobreva, Arturo Oncevay, Antonio Valerio Miceli Barone, and Philip Williams. 2020a. The university of edinburgh's english-tamil and english-inuktitut submissions to the wmt20 news translation task. In *Proceedings of the Fifth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Rachel Bawden, Giorgio Maria Di Nunzio, Cristian Grozea, Inigo Jauregi Unanue, Antonio Jimeno Yepes, Nancy Mah, David Martinez, Aurélie Névéol, Mariana Neves, Maite Oronoz, Olatz Perez-de Viñaspre, Massimo Piccardi, Roland Roller, Amy Siu, Philippe Thomas, Federica Vezzani, Maika Vicente Navarro, Dina Wiemann, and Lana Yeganova. 2020b. Findings of the wmt 2020 biomedical translation shared task: Basque, italian and russian as new additional languages. In *Proceedings of the Fifth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Chao Bei, Hao Zong, Qingmin Liu, and Conghu Yuan. 2020. Gtcom neural machine translation systems for wmt20. In *Proceedings of the Fifth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Pushpak Bhattacharyya. 2010. IndoWordNet. In *Proceedings of LREC*.

Ondřej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Aleš Tamchyna. 2014. Findings of the 2014 Workshop on

Statistical Machine Translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 12–58, Baltimore, Maryland, USA. Association for Computational Linguistics.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Raphael Rubino, Lucia Specia, and Marco Turchi. 2017. Findings of the 2017 Conference on Machine Translation (WMT17). In *Proceedings of the Second Conference on Machine Translation*, Copenhagen, Denmark. Association for Computational Linguistics.

Ondřej Bojar, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Philipp Koehn, and Christof Monz. 2018. Findings of the 2018 conference on machine translation (WMT18). In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 272–303, Belgium, Brussels. Association for Computational Linguistics.

Ondřej Bojar, Christian Buck, Chris Callison-Burch, Christian Federmann, Barry Haddow, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. 2013. Findings of the 2013 Workshop on Statistical Machine Translation. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 1–44, Sofia, Bulgaria. Association for Computational Linguistics.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurelie Neveol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. 2016. Findings of the 2016 Conference on Machine Translation. In *Proceedings of the First Conference on Machine Translation*, pages 131–198, Berlin, Germany. Association for Computational Linguistics.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Barry Haddow, Matthias Huck, Chris Hokamp, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Carolina Scarton, Lucia Specia, and Marco Turchi. 2015. Findings of the 2015 Workshop on Statistical Machine Translation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 1–46, Lisbon, Portugal. Association for Computational Linguistics.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, André Martins, Christof Monz, Matteo Negri, Aurélie Névéol, Mariana Neves, Matt Post, Marco Turchi, and Karin Verspoor, editors. 2019. *Proceedings of the Fourth Conference on Machine Translation*. Association for Computational Linguistics, Florence, Italy.

Pere Vergés Boncompte and Marta R. Costa-jussà. 2020. Multilingual neural machine translation: Case-study for catalan, spanish and portuguese romance languages. In *Proceedings of WMT*.

Chris Callison-Burch, Cameron Shaw Fordyce, Philipp Koehn, Christof Monz, and Josh Schroeder. 2007. (Meta-) Evaluation of Machine Translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 136–158, Prague, Czech Republic. Association for Computational Linguistics.

Chris Callison-Burch, Cameron Shaw Fordyce, Philipp Koehn, Christof Monz, and Josh Schroeder. 2008. Further Meta-Evaluation of Machine Translation. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 70–106, Columbus, Ohio. Association for Computational Linguistics.

Chris Callison-Burch, Philipp Koehn, Christof Monz, Kay Peterson, Mark Przybocki, and Omar Zaidan. 2010. Findings of the 2010 Joint Workshop on Statistical Machine Translation and Metrics for Machine Translation. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 17–53, Uppsala, Sweden. Association for Computational Linguistics.

Chris Callison-Burch, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. 2012. Findings of the 2012 Workshop on Statistical Machine Translation. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 10–48, Montreal, Canada. Association for Computational Linguistics.

Chris Callison-Burch, Philipp Koehn, Christof Monz, and Josh Schroeder. 2009. Findings of the 2009 Workshop on Statistical Machine Translation. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 1–28, Athens, Greece. Association for Computational Linguistics.

Chris Callison-Burch, Philipp Koehn, Christof Monz, and Omar Zaidan. 2011. Findings of the 2011 Workshop on Statistical Machine Translation. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 22–64, Edinburgh, Scotland. Association for Computational Linguistics.

Sheila Castilho, Maja Popović, and Andy Way. 2020. On context span needed for machine translation evaluation. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 3735–3742, Marseille, France. European Language Resources Association.

Mauro Cettolo, Christian Girardi, and Marcello Federico. 2012. WIT3: web inventory of transcribed and translated talks. In *Proc. of EAMT*, pages 261–268.

Rajen Chatterjee, Markus Freitag, Matteo Negri, and Marco Turchi. 2020. Findings of the wmt 2020

shared task on automatic post-editing. In *Proceedings of the Fifth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Peng-Jen Chen, Ann Lee, Changhan Wang, Naman Goyal, Angela Fan, Mary Williamson, and Jiatao Gu. 2020a. Facebook ai's wmt20 news translation task submission. In *Proceedings of the Fifth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Tanfang Chen, Weiwei Wang, Wenyang Wei, Xing Shi, Xiangang Li, Jieping Ye, and Kevin Knight. 2020b. The didi machine translation system for wmt20. In *Proceedings of the Fifth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 7059–7069. Curran Associates, Inc.

Marta R. Costa-jussà, Roger Creus, Oriol Domingo, Albert Domínguez, Miquel Escobar, Cayetana López, Marina Garcia, and Margarita Geleta. 2020. MT-Adapted Datasheets for Datasets: Template and Repository. *arXiv e-prints*, page arXiv:2005.13156.

Marta R. Costa-jussà, Marcos Zampieri, and Santanu Pal. 2018. A Neural Approach to Language Variety Translation. In *Proceedings of VarDial*.

Farhan Dhanani and Muhammad Rafi. 2020. Attention transformer model for translation of similar languages. In *Proceedings of WMT*.

Prajit Dhar, Arianna Bisazza, and Gertjan van Noord. 2020. Linguistically motivated subwords improve english-tamil translation: University of groningen's submission to wmt-2020. In *Proceedings of the Fifth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Tomaž Erjavec, Nikola Ljubešić, and Nataša Logar. 2015. The slwac corpus of the sloveneweb. *Informatica*, 39(1).

Carlos Escolano, Marta R. Costa-jussà, and José A. R. Fonollosa. 2020. The talp-upc machine translation systems for wmt20 news translation task: Multilingual adaptation for low resource mt. In *Proceedings of the Fifth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Miquel Esplà-Gomis. 2009. Bitextor: a free/open-source software to harvest translation memories from multilingual websites. In *MT Summit Workshop on New Tools for Translators*. International Association for Machine Translation.

M. Amin Farajian, António V. Lopes, André F. T. Martins, Sameen Maruf, and Gholamreza Haffari. 2020. Findings of the wmt 2020 shared task on chat translation. In *Proceedings of the Fifth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Christian Federmann. 2012. Appraise: an open-source toolkit for manual evaluation of mt output. *The Prague Bulletin of Mathematical Linguistics*, 98:25–35.

Mikel L. Forcada, Francis M. Tyers, and Gema Ramírez Sánchez. 2009-11. The apertium machine translation platform: five years on.

Alexander Fraser. 2020. The wmt 2020 shared tasks in unsupervised mt and very low resource supervised mt. In *Proceedings of the Fifth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Markus Freitag, Isaac Caswell, and Scott Roy. 2019. APE at scale and its implications on MT evaluation biases. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 34–44, Florence, Italy. Association for Computational Linguistics.

Markus Freitag, David Grangier, and Isaac Caswell. 2020. BLEU might be guilty but references are not innocent. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 61–71, Online. Association for Computational Linguistics.

Ulrich Germann. 2020. The university of edinburgh's submission to the german-to-english and english-to-german tracks in the wmt 2020 news translation and zero-shot translation robustness tasks. In *Proceedings of the Fifth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Ulrich Germann, Roman Grundkiewicz, Martin Popel, Radina Dobreva, Nikolay Bogoychev, and Kenneth Heafield. 2020. Speed-optimized, compact student models that distill knowledge from a larger teacher model: the uedin-cuni submission to the wmt 2020 news translation task. In *Proceedings of the Fifth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Vikrant Goyal, Anoop Kunchukuttan, Rahul Kejriwal, Siddharth Jain, and Amit Bhagwat. 2020. Contact relatedness can help improve multilingual nmt: Microsoft stci-mt @ wmt20. In *Proceedings of the Fifth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Yvette Graham, George Awad, and Alan Smeaton. 2018. Evaluation of automatic video captioning using direct assessment. *PLOS ONE*, 13(9):1–20.

Yvette Graham, Timothy Baldwin, Alistair Moffat, and Justin Zobel. 2013. Continuous Measurement Scales in Human Evaluation of Machine Translation. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*,

pages 33–41, Sofia, Bulgaria. Association for Computational Linguistics.

Yvette Graham, Timothy Baldwin, Alistair Moffat, and Justin Zobel. 2014. Is Machine Translation Getting Better over Time? In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 443–451, Gothenburg, Sweden. Association for Computational Linguistics.

Yvette Graham, Timothy Baldwin, Alistair Moffat, and Justin Zobel. 2016. Can machine translation systems be evaluated by the crowd alone. *Natural Language Engineering*, pages 1–28.

Yvette Graham, Barry Haddow, and Philipp Koehn. 2019. Translationese in Machine Translation Evaluation. *arXiv e-prints*, page arXiv:1906.09833.

Yvette Graham, Barry Haddow, and Philipp Koehn. 2020. Statistical power and translationese in machine translation evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, Virtual.

Francisco Guzmán, Peng-Jen Chen, Myle Ott, Juan Pino, Guillaume Lample, Philipp Koehn, Vishrav Chaudhary, and Marc'Aurelio Ranzato. 2019. The FLORES evaluation datasets for low-resource machine translation: Nepali–English and Sinhala–English. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6098–6111, Hong Kong, China. Association for Computational Linguistics.

Jeremy Gwinnup and Tim Anderson. 2020. The afrl wmt20 news-translation systems. In *Proceedings of the Fifth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Barry Haddow and Faheem Kirefu. 2020. PMIndia – A Collection of Parallel Corpora of Languages of India. *arXiv e-prints*, page arXiv:2001.09907.

Sami Ul Haq, Sadaf Abdul Rauf, Arsalan Shaukat, and Abdullah Saeed. 2020. Document level nmt of low-resource languages with backtranslation. In *Proceedings of WMT*.

Hossein Hassani. 2017. Kurdish Interdialect Machine Translation. *Proceedings of VarDial*.

François Hernandez and Vincent Nguyen. 2020. The ubiqus english-inuktiut system for wmt20. In *Proceedings of the Fifth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Hideki Isozaki, Tsutomu Hirao, Kevin Duh, Katsuhito Sudoh, and Hajime Tsukada. 2010. Automatic Evaluation of Translation Quality for Distant Language Pairs. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 944–952, Cambridge, MA. Association for Computational Linguistics.

Eric Joanis, Rebecca Knowles, Roland Kuhn, Samuel Larkin, Patrick Littell, Chi-kiu Lo, Darlene Stewart, and Jeffrey Micher. 2020. The Nunavut Hansard Inuktitut–English parallel corpus 3.0 with preliminary machine translation results. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 2562–2572, Marseille, France. European Language Resources Association.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Jiwan Kim, Soyoon Park, Sangha Kim, and Yoonjung Choi. 2020. An iterative knowledge transfer nmt system for wmt20 news translation task. In *Proceedings of the Fifth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Shun Kiyono, Takumi Ito, Ryuto Konno, Makoto Morishita, and Jun Suzuki. 2020. Tohoku-aip-ntt at wmt 2020 news translation task. In *Proceedings of the Fifth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Rebecca Knowles, Darlene Stewart, Samuel Larkin, and Patrick Littell. 2020. Nrc systems for the 2020 inuktitut-english news translation task. In *Proceedings of the Fifth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Tom Kocmi. 2020. Cuni submission for inuktitut language in wmt news 2020. In *Proceedings of the Fifth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Tom Kocmi, Tomasz Limisiewicz, and Gabriel Stanovsky. 2020. Gender coreference and bias evaluation at wmt 2020. In *Proceedings of the Fifth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Philipp Koehn, Vishrav Chaudhary, Ahmed El-Kishky, Naman Goyal, Peng-Jen Chen, and Francisco Guzmán. 2020. Findings of the wmt 2020 shared task on parallel corpus filtering and alignment. In *Proceedings of the Fifth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Christopher J. Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the 45th*

Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Philipp Koehn and Christof Monz. 2006. Manual and Automatic Evaluation of Machine Translation between European Languages. In *Proceedings on the Workshop on Statistical Machine Translation*, pages 102–121, New York City. Association for Computational Linguistics.

Rihards Krišlauks and Mārcis Pinnis. 2020. Tilde at wmt 2020: News task systems. In *Proceedings of the Fifth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Mateusz Krubiński, Marcin Chochowski, Bartłomiej Boczek, Mikołaj Koszowski, Adam Dobrowolski, Marcin Szymański, and Paweł Przybysz. 2020. Samsung r&d institute poland submission to wmt20 news translation task. In *Proceedings of the Fifth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Amit Kumar, Rupjyoti Baruah, Rajesh Kumar Mundotiya, and Anil Kumar Singh. 2020. Transformer-based neural machine translation system for hindi - marathi. In *Proceedings of WMT*.

Anoop Kunchukuttan. 2020a. Indowordnet parallel corpus.

Anoop Kunchukuttan. 2020b. The IndicNLP Library. https://github.com/anoopkunchukuttan/indic_nlp_library/blob/master/docs/indicnlp.pdf.

Anoop Kunchukuttan, Pratik Mehta, and Pushpak Bhattacharyya. 2018. The IIT Bombay English-Hindi parallel corpus. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Sahinur Rahman Laskar, Abdullah Faiz Ur Rahman Khilji, Partha Pakray, and Sivaji Bandyopadhyay. 2020. Hindi-marathi cross lingual model. In *Proceedings of WMT*.

Samuel Laubli, Sheila Castilho, Graham Neubig, Rico Sennrich, Qinlan Shen, and Antonio Toral. 2020. A set of recommendations for assessing human–machine parity in language translation. *Journal of Artificial Intelligence Research (JAIR)*, 67.

Samuel Läubli, Rico Sennrich, and Martin Volk. 2018. Has Neural Machine Translation Achieved Human Parity? A Case for Document-level Evaluation. In *EMNLP 2018*, Brussels, Belgium. Association for Computational Linguistics.

Zuchao Li, Hai Zhao, Rui Wang, Kehai Chen, Masao Utiyama, and Eiichiro Sumita. 2020. Sjtu-nict's supervised and unsupervised neural machine translation systems for the wmt20 news translation task. In *Proceedings of the Fifth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Zihan Liu, Genta Indra Winata, Zhaojiang Lin, Peng Xu, and Pascale Fung. 2019. Attention-informed mixed-language training for zero-shot cross-lingual task-oriented dialogue systems.

Nikola Ljubešić and Tomaž Erjavec. 2011. hrwac and slwac: Compiling web corpora for croatian and slovene. In *International Conference on Text, Speech and Dialogue*, pages 395–402. Springer.

Nikola Ljubešić and Filip Klubička. 2014. {bs, hr, sr} wac-web corpora of bosnian, croatian and serbian. In *Proceedings of the 9th web as corpus workshop (WaC-9)*, pages 29–35.

Qingsong Ma, Yvette Graham, Timothy Baldwin, and Qun Liu. 2017. Further investigation into reference bias in monolingual evaluation of machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2466–2475, Copenhagen, Denmark. Association for Computational Linguistics.

Vivien Macketanz, Eleftherios Avramidis, Aljoscha Burchardt, and Hans Uszkoreit. 2018. Fine-grained evaluation of German-English Machine Translation based on a Test Suite. In *Proceedings of the Third Conference on Machine Translation*, Brussels, Belgium. Association for Computational Linguistics.

Lovish Madaan, Soumya Sharma, and Parag Singla. 2020. Transfer learning for related languages: Iit delhi's submissions to the wmt20 similar language translation task. In *Proceedings of WMT*.

Benjamin Marie, Raphael Rubino, and Atsushi Fujita. 2020. Combination of neural machine translation systems at wmt20. In *Proceedings of the Fifth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Luıs Marujo, Nuno Grazina, Tiago Luıs, Wang Ling, Luısa Coheur, and Isabel Trancoso. 2011. BP2EP–Adaptation of Brazilian Portuguese texts to European Portuguese. In *Proceedings of EAMT*.

Nitika Mathur, Johnny Wei, Markus Freitag, Qingsong Ma, and Ondřej Bojar. 2020. Results of the wmt20 metrics shared task. In *Proceedings of the Fifth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Fandong Meng, Jianhao Yan, Yijin Liu, Yuan Gao, Xianfeng Zeng, Qinsong Zeng, Peng Li, Ming Chen, Jie Zhou, Sifan Liu, and Hao Zhou. 2020. Wechat neural machine translation systems for wmt20. In *Proceedings of the Fifth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Luis A. Menéndez-Salazar, Grigori Sidorov, and Marta R. Costa-Jussà. 2020. The ipn-cic team system submission for the wmt 2020 simmilar language task. In *Proceedings of WMT*.

Simon Mille, Anja Belz, Bernd Bohnet, Yvette Graham, Emily Pitler, and Leo Wanner. 2018. The first multilingual surface realisation shared task (sr'18): Overview and evaluation results. In *Proceedings of the First Workshop on Multilingual Surface Realisation*, pages 1–12. Association for Computational Linguistics.

Simon Mille, Anja Belz, Bernd Bohnet, Yvette Graham, and Leo Wanner. 2019. The second multilingual surface realisation shared task (SR'19): Overview and evaluation results. In *Proceedings of the 2nd Workshop on Multilingual Surface Realisation (MSR 2019)*, pages 1–17, Hong Kong, China. Association for Computational Linguistics.

Alexander Molchanov. 2020. Promt systems for wmt 2020 shared news translation task. In *Proceedings of the Fifth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Makoto Morishita, Jun Suzuki, and Masaaki Nagata. 2020. JParaCrawl: A large scale web-based English-Japanese parallel corpus. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 3603–3609.

Vandan Mujadia and Dipti Sharmaa. 2020. Nmt based similar language translation for hindi - marathi. In *Proceedings of WMT*.

Graham Neubig. 2011. The Kyoto free translation task. http://www.phontron.com/kftt.

Atul Kr. Ojha, Priya Rani, Akanksha Bansal, Bharathi Raja Chakravarthi, Ritesh Kumar, and John P. McCrae. 2020. Nuig-panlingua-kmi hindi–marathi mt systems for similar language translation task @ wmt 2020. In *Proceedings of WMT*.

Csaba Oravecz, Katina Bontcheva, László Tihanyi, David Kolovratnik, Bhavani Bhaskar, Adrien Lardilleux, Szymon Klocek, and Andreas Eisele. 2020. etranslation's submissions to the wmt 2020 news translation task. In *Proceedings of the Fifth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. 2018. Scaling Neural Machine Translation. In *Proceedings of the Third Conference on Machine Translation*, Brussels, Belgium. Association for Computational Linguistics.

Santanu Pal and Marcos Zampieri. 2020. Neural machine translation for similar languages: The case of indo-aryan languages. In *Proceedings of WMT*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318, Philadelphia, Pennsylvania. Association for Computational Linguistics.

Venkatesh Parthasarathy, Akshai Ramesh, Rejwanul Haque, and Andy Way. 2020. The adapt system description for the wmt20 news translation task. In *Proceedings of the Fifth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Ellie Pavlick, Matt Post, Ann Irvine, Dmitry Kachaev, and Chris Callison-Burch. 2014. The language demographics of Amazon Mechanical Turk. *Transactions of the Association for Computational Linguistics*, 2(Feb):79–92.

Martin Popel. 2018. CUNI transformer neural MT system for WMT18. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 482–487, Belgium, Brussels. Association for Computational Linguistics.

Martin Popel. 2020. Cuni english-czech and english-polish systems in wmt20: Robust document-level training. In *Proceedings of the Fifth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Martin Popel and Ondřej Bojar. 2018. Training Tips for the Transformer Model. *The Prague Bulletin of Mathematical Linguistics*, 110(1):43–70.

Martin Popel, Marketa Tomkova, Jakub Tomek, Łukasz Kaiser, Jakob Uszkoreit, Ondřej Bojar, and Zdeněk Žabokrtský. 2020. Transforming machine translation: a deep learning system reaches news translation quality comparable to human professionals. *Nature Communications*, 11(4381):1–15.

Maja Popovic. 2019. On reducing translation shifts in translations intended for MT evaluation. In *Proceedings of Machine Translation Summit XVII Volume 2: Translator, Project and User Tracks*, pages 80–87, Dublin, Ireland. European Association for Machine Translation.

Maja Popović, Alberto Poncelas, Marija Brkic, and Andy Way. 2020. Neural machine translation for translating into Croatian and Serbian. In *Proceedings of the Seventh Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*.

Maja Popović and Alberto Poncelas. 2020. Neural machine translation between similar south-slavic languages. In *Proceedings of WMT*.

R. Pryzant, Y. Chung, D. Jurafsky, and D. Britz. 2017. JESC: Japanese-English Subtitle Corpus. *arXiv preprint arXiv:1710.10639*.

Alessandro Raganato, Yves Scherrer, and Jörg Tiedemann. 2019. The MuCoW Test Suite at WMT 2019: Automatically Harvested Multilingual Contrastive Word Sense Disambiguation Test Sets for Machine

Translation. In *Proceedings of the Fourth Conference on Machine Translation*, Florence, Italy. Association for Computational Linguistics.

Kamalkumar Rathinasamy, Amanpreet Singh, Balaguru Sivasambagupta, Prajna Prasad Neerchal, and Vani Sivasankaran. 2020. Infosys machine translation system for wmt20 similar language translation task. In *Proceedings of WMT*.

Christian Roest, Lukas Edman, Gosse Minnema, Kevin Kelly, Jennifer Spenader, and Antonio Toral. 2020. Machine translation for english–inuktitut with segmentation,data acquisition and pre-training. In *Proceedings of the Fifth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Yves Scherrer, Stig-Arne Grönroos, and Sami Virpioja. 2020a. The university of helsinki and aalto university submissions to the wmt 2020 news and low-resource translation tasks. In *Proceedings of the Fifth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Yves Scherrer, Alessandro Raganato, and Jörg Tiedemann. 2020b. The mucow word sense disambiguation test suite at wmt 2020. In *Proceedings of the Fifth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán. 2019. WikiMatrix: Mining 135M Parallel Sentences in 1620 Language Pairs from Wikipedia. *arXiv e-prints*, page arXiv:1907.05791.

Tingxun Shi, Shiyu Zhao, Xiaopu Li, Xiaoxue Wang, Qian Zhang, Di Ai, Dawei Dang, Xue Zhengshan, and JIE HAO. 2020. Oppo's machine translation systems for wmt20. In *Proceedings of the Fifth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Shashank Siripragada, Jerin Philip, Vinay P. Namboodiri, and C V Jawahar. 2020. A multilingual parallel corpora collection effort for Indian languages. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 3743–3751, Marseille, France. European Language Resources Association.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and Ralph Weischedel. 2006. A study of translation error rate with targeted human annotation. In *In Proceedings of the Association for Machine Transaltion in the Americas (AMTA 2006*.

Lucia Specia, Frédéric Blain, Marina Fomicheva, Erick Fonseca, Vishrav Chaudhary, Francisco Guzmán, and André F. T. Martins. 2020a. Findings of the wmt 2020 shared task on quality estimation. In *Proceedings of the Fifth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Lucia Specia, Zhenhao Li, Juan Pino, Vishrav Chaudhary, Francisco Guzmán, Graham Neubig, Nadir Durrani, Yonatan Belinkov, Philipp Koehn, Hassan Sajjad, Paul Michel, and Xian Li. 2020b. Findings of the wmt 2020 shared task on machine translation robustness. In *Proceedings of the Fifth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Gabriel Stanovsky, Noah A. Smith, and Luke Zettlemoyer. 2019. Evaluating gender bias in machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1679–1684, Florence, Italy. Association for Computational Linguistics.

Roman Sudarikov, Martin Popel, Ondřej Bojar, Aljoscha Burchardt, and Ondřej Klejch. 2016. Using MT-ComparEval. In *Translation Evaluation: From Fragmented Tools and Data Sets to an Integrated Ecosystem*, pages 76–82.

Jörg Tiedemann. 2009. News from OPUS - a collection of multilingual parallel corpora with tools and interfaces. In *Proceedings of the Conference on Recent Advances in Natural Language Processing*, pages 237–248. John Benjamins.

Jörg Tiedemann and Lars Nygaard. 2004. The opus corpus-parallel and free: http://logos.uio.no/opus. In *Proceedings of LREC*.

Antonio Toral, Sheila Castilho, Ke Hu, and Andy Way. 2018. Attaining the Unattainable? Reassessing Claims of Human Parity in Neural Machine Translation. In *Proceedings of the Third Conference on Machine Translation*, Brussels, Belgium. Association for Computational Linguistics.

Dániel Varga, Péter Halaácsy, András Kornai, Voktor Nagy, László Németh, and Viktor Trón. 2005. Parallel corpora for medium density languages. In *Proceedings of the RANLP 2005 Conference*, pages 590–596.

Tereza Vojtěchová, Michal Novák, Miloš Klouček, and Ondřej Bojar. 2019. SAO WMT19 Test Suite: Machine Translation of Audit Reports. In *Proceedings of the Fourth Conference on Machine Translation*, Florence, Italy. Association for Computational Linguistics.

Daimeng Wei, Hengchao Shang, Zhanglin Wu, Zhengzhe Yu, Liangyou Li, Jiaxin Guo, Minghan Wang, Hao Yang, Lizhi Lei, Ying Qin, and Shiliang Sun. 2020a. Hw-tsc's participation in the wmt 2020 news translation shared task. In *Proceedings of the Fifth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Xiangpeng Wei, Ping Guo, Yunpeng Li, Xingsheng Zhang, Luxi Xing, and Yue Hu. 2020b. Iie's neural machine translation systems for wmt20. In *Proceedings of the Fifth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Liwei Wu, Xiao Pan, Zehui Lin, Yaoming ZHU, Mingxuan Wang, and Lei Li. 2020a. The volctrans machine translation system for wmt20. In *Proceedings of the Fifth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Shuangzhi Wu, Xing Wang, Longyue Wang, Fangxu Liu, Jun Xie, Zhaopeng Tu, Shuming Shi, and Mu Li. 2020b. Tencent neural machine translation systems for the wmt20 news translation task. In *Proceedings of the Fifth Conference on Machine Translation*, Online. Association for Computational Linguistics.

ariel Xv. 2020. Russian-english bidirectional machine translation system. In *Proceedings of the Fifth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Saumitra Yadav and Manish Shrivastava. 2020. A3-108 machine translation system for similar language translation shared task 2020. In *Proceedings of WMT*.

Lei Yu, Laurent Sartran, Po-Sen Huang, Wojciech Stokowiec, Domenic Donato, Srivatsan Srinivasan, Alek Andreev, Wang Ling, Sona Mokra, Agustin Dal Lago, Yotam Doron, Susannah Young, Phil Blunsom, and Chris Dyer. 2020. The deepmind chinese–english document translation system at wmt2020. In *Proceedings of the Fifth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Jiacheng Zhang, Yanzhuo Ding, Shiqi Shen, Yong Cheng, Maosong Sun, Huan-Bo Luan, and Yang Liu. 2017. THUMT: an open source toolkit for neural machine translation. *CoRR*, abs/1706.06415.

Xiaoheng Zhang. 1998. Dialect MT: A Case Study Between Cantonese and Mandarin. In *Proceedings of ACL*.

Yuhao Zhang, Ziyang Wang, Runzhe Cao, Binghao Wei, Weiqiao Shan, Shuhan Zhou, Abudurexiti Reheman, Tao Zhou, Xin Zeng, Laohu Wang, Yongyu Mu, Jingnan Zhang, Xiaoqian Liu, Xuanjun Zhou, Yinqiao Li, Bei Li, Tong Xiao, and Jingbo Zhu. 2020. The niutrans machine translation systems for wmt20. In *Proceedings of the Fifth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Vilém Zouhar, Tereza Vojtěchová, and Ondřej Bojar. 2020. Wmt20 document-level markable error exploration. In *Proceedings of the Fifth Conference on Machine Translation*, Online. Association for Computational Linguistics.

A Differences in Human Scores

Tables 33–50 show differences in average standardized human scores for all pairs of competing systems for each language pair. The numbers in each of the tables' cells indicate the difference in average standardized human scores for the system in that column and the system in that row.

Because there were so many systems and data conditions the significance of each pairwise comparison needs to be quantified. We applied Wilcoxon rank-sum test to measure the likelihood that such differences could occur simply by chance. In the following tables $\star$ indicates statistical significance at $p < 0.05$, † indicates statistical significance at $p < 0.01$, and ‡ indicates statistical significance at $p < 0.001$, according to Wilcoxon rank-sum test.

Each table contains final rows showing the average score achieved by that system and the rank range according according to Wilcoxon rank-sum test ($p < 0.05$). Gray lines separate clusters based on non-overlapping rank ranges.

	CUNI-DocTransformer	OPPO	Online-B	CUNI-Transformer	Online-A	SRPOL	UEDIN-CUNI	CUNI-T2T-2018	Online-G	Online-Z	PROMT-NMT	zlabs-nlp
CUNI-DocTransformer	-	0.05	0.08	0.08⋆	0.10	0.15†	0.17‡	0.18‡	0.19‡	0.20‡	0.21‡	0.26‡
OPPO	-0.05	-	0.03	0.04	0.05	0.11⋆	0.12‡	0.14‡	0.14‡	0.15‡	0.17‡	0.21‡
Online-B	-0.08	-0.03	-	0.01	0.02	0.08	0.09†	0.11†	0.11†	0.12†	0.14†	0.18‡
CUNI-Transformer	-0.08	-0.04	-0.01	-	0.02	0.07	0.08⋆	0.10⋆	0.10†	0.11†	0.13†	0.18‡
Online-A	-0.10	-0.05	-0.02	-0.02	-	0.05	0.07⋆	0.08⋆	0.09⋆	0.10†	0.11†	0.16‡
SRPOL	-0.15	-0.11	-0.08	-0.07	-0.05	-	0.01	0.03	0.03	0.04	0.06	0.10†
UEDIN-CUNI	-0.17	-0.12	-0.09	-0.08	-0.07	-0.01	-	0.02	0.02	0.03	0.05	0.09⋆
CUNI-T2T-2018	-0.18	-0.14	-0.11	-0.10	-0.08	-0.03	-0.02	-	0.00	0.02	0.03	0.08
Online-G	-0.19	-0.14	-0.11	-0.10	-0.09	-0.03	-0.02	0.00	-	0.01	0.03	0.07
Online-Z	-0.20	-0.15	-0.12	-0.11	-0.10	-0.04	-0.03	-0.02	-0.01	-	0.01	0.06
PROMT-NMT	-0.21	-0.17	-0.14	-0.13	-0.11	-0.06	-0.05	-0.03	-0.03	-0.01	-	0.05
zlabs-nlp	-0.26	-0.21	-0.18	-0.18	-0.16	-0.10	-0.09	-0.08	-0.07	-0.06	-0.05	-
score	0.12	0.07	0.04	0.03	0.02	-0.04	-0.05	-0.07	-0.07	-0.08	-0.09	-0.14
rank	1–12	1–12	1–12	1–12	1–12	1–12	1–12	1–12	1–12	1–12	1–12	1–12

Table 33: Head to head comparison for Czech→English systems

	VolcTrans	DiDi-NLP	WeChat-AI	Tencent-Translation	Online-B	DeepMind	OPPO	THUNLP	SJTU-NICT	Huawei-TSC	Online-A	HUMAN	Online-G	dong-NMT	zlabs-NLP	Online-Z	WMTBiomedBaseline
VolcTrans	-	0.01⋆	0.02†	0.04†	0.04⋆	0.05†	0.05‡	0.07†	0.09‡	0.10‡	0.12‡	0.13‡	0.17‡	0.18‡	0.21‡	0.24‡	0.43‡
DiDi-NLP	-0.01	-	0.01	0.03	0.03	0.04	0.04⋆	0.06	0.07	0.09‡	0.11†	0.12‡	0.16‡	0.17‡	0.19‡	0.22‡	0.42‡
WeChat-AI	-0.02	-0.01	-	0.01	0.02	0.03	0.03	0.05	0.06	0.08†	0.09⋆	0.11‡	0.15‡	0.16‡	0.18‡	0.21‡	0.41‡
Tencent-Translation	-0.04	-0.03	-0.01	-	0.00	0.01	0.01	0.04	0.05	0.06⋆	0.08	0.09†	0.13‡	0.14‡	0.17‡	0.20‡	0.40‡
Online-B	-0.04	-0.03	-0.02	0.00	-	0.01	0.01	0.03	0.04	0.06†	0.08⋆	0.09‡	0.13‡	0.14‡	0.17‡	0.20‡	0.39‡
DeepMind	-0.05	-0.04	-0.03	-0.01	-0.01	-	0.00	0.02	0.03	0.05†	0.07⋆	0.08†	0.12‡	0.13‡	0.16‡	0.19‡	0.38‡
OPPO	-0.05	-0.04	-0.03	-0.01	-0.01	0.00	-	0.02	0.03	0.05⋆	0.07	0.08†	0.12‡	0.13‡	0.16‡	0.19‡	0.38‡
THUNLP	-0.07	-0.06	-0.05	-0.04	-0.03	-0.02	-0.02	-	0.01	0.03†	0.04	0.06†	0.10‡	0.11‡	0.13‡	0.16‡	0.36‡
SJTU-NICT	-0.09	-0.07	-0.06	-0.05	-0.04	-0.03	-0.03	-0.01	-	0.02⋆	0.03	0.05†	0.09‡	0.09‡	0.12‡	0.15‡	0.35‡
Huawei-TSC	-0.10	-0.09	-0.08	-0.06	-0.06	-0.05	-0.05	-0.03	-0.02	-	0.02	0.03	0.07⋆	0.08†	0.11†	0.13‡	0.33‡
Online-A	-0.12	-0.11	-0.09	-0.08	-0.08	-0.07	-0.07	-0.04	-0.03	-0.02	-	0.01	0.05†	0.06‡	0.09‡	0.12‡	0.32‡
HUMAN	-0.13	-0.12	-0.11	-0.09	-0.09	-0.08	-0.08	-0.06	-0.05	-0.03	-0.01	-	0.04	0.05⋆	0.08†	0.11‡	0.30‡
Online-G	-0.17	-0.16	-0.15	-0.13	-0.13	-0.12	-0.12	-0.10	-0.09	-0.07	-0.05	-0.04	-	0.01	0.03	0.06⋆	0.26‡
dong-NMT	-0.18	-0.17	-0.16	-0.14	-0.14	-0.13	-0.13	-0.11	-0.09	-0.08	-0.06	-0.05	-0.01	-	0.03	0.06	0.25‡
zlabs-NLP	-0.21	-0.19	-0.18	-0.17	-0.17	-0.16	-0.16	-0.13	-0.12	-0.11	-0.09	-0.08	-0.03	-0.03	-	0.03	0.23‡
Online-Z	-0.24	-0.22	-0.21	-0.20	-0.20	-0.19	-0.19	-0.16	-0.15	-0.13	-0.12	-0.11	-0.06	-0.06	-0.03	-	0.20‡
WMTBiomedBaseline	-0.43	-0.42	-0.41	-0.40	-0.39	-0.38	-0.38	-0.36	-0.35	-0.33	-0.32	-0.30	-0.26	-0.25	-0.23	-0.20	-
score	0.10	0.09	0.08	0.06	0.06	0.05	0.05	0.03	0.02	0.00	-0.02	-0.03	-0.07	-0.08	-0.11	-0.14	-0.33
rank	1	2–16	2–16	2–16	2–16	2–16	2–16	2–16	2–16	2–16	2–16	2–16	2–16	2–16	2–16	2–16	17

Table 34: Head to head comparison for Chinese→English systems

	VolcTrans	OPPO	HUMAN	Tohoku-AIP-NTT	Online-A	Online-G	PROMT-NMT	Online-B	UEDIN	Online-Z	WMTBiomedBaseline	zlabs-NLP	yolo
VolcTrans	-	0.01	0.04⋆	0.05	0.05	0.06	0.06⋆	0.06⋆	0.10⋆	0.14‡	0.31‡	0.33‡	1.85‡
OPPO	-0.01	-	0.03⋆	0.04	0.04	0.05	0.05⋆	0.05⋆	0.09	0.13‡	0.30‡	0.33‡	1.84‡
HUMAN	-0.04	-0.03	-	0.01	0.01	0.01	0.02	0.02	0.06	0.10	0.27‡	0.29‡	1.80‡
Tohoku-AIP-NTT	-0.05	-0.04	-0.01	-	0.00	0.01	0.01	0.01	0.05	0.09†	0.26‡	0.28‡	1.80‡
Online-A	-0.05	-0.04	-0.01	0.00	-	0.01	0.01	0.01	0.05	0.09†	0.26‡	0.28‡	1.80‡
Online-G	-0.06	-0.05	-0.01	-0.01	-0.01	-	0.00	0.01	0.04	0.09†	0.25‡	0.28‡	1.79‡
PROMT-NMT	-0.06	-0.05	-0.02	-0.01	-0.01	0.00	-	0.00	0.04	0.09	0.25‡	0.28‡	1.79‡
Online-B	-0.06	-0.05	-0.02	-0.01	-0.01	-0.01	0.00	-	0.04	0.08	0.25‡	0.27‡	1.78‡
UEDIN	-0.10	-0.09	-0.06	-0.05	-0.05	-0.04	-0.04	-0.04	-	0.05⋆	0.21‡	0.24‡	1.75‡
Online-Z	-0.14	-0.13	-0.10	-0.09	-0.09	-0.09	-0.09	-0.08	-0.05	-	0.16‡	0.19‡	1.70‡
WMTBiomedBaseline	-0.31	-0.30	-0.27	-0.26	-0.26	-0.25	-0.25	-0.25	-0.21	-0.16	-	0.03	1.54‡
zlabs-NLP	-0.33	-0.33	-0.29	-0.28	-0.28	-0.28	-0.28	-0.27	-0.24	-0.19	-0.03	-	1.51‡
yolo	-1.85	-1.84	-1.80	-1.80	-1.80	-1.79	-1.79	-1.78	-1.75	-1.70	-1.54	-1.51	-
score	0.23	0.22	0.19	0.18	0.18	0.17	0.17	0.17	0.13	0.09	-0.08	-0.11	-1.62
rank	1–9	1–9	1–9	1–9	1–9	1–9	1–9	1–9	1–9	10	11–12	11–12	13

Table 35: Head to head comparison for German→English systems

	ONLINE-G	ONLINE-A	OPPO	eTRANSLATION	PROMT-NMT	ONLINE-B	HUMAN	ARIEL XV	AFRL	DIDI-NLP	ONLINE-Z	ZLABS-NLP
ONLINE-G	-	0.01	0.01	0.02	0.03	0.05†	0.06	0.08⋆	0.10†	0.14‡	0.15‡	0.28‡
ONLINE-A	-0.01	-	0.00	0.01	0.02	0.04	0.05	0.07	0.09⋆	0.13†	0.14†	0.27‡
OPPO	-0.01	0.00	-	0.01	0.02	0.04⋆	0.05	0.07⋆	0.09†	0.13†	0.13‡	0.27‡
eTRANSLATION	-0.02	-0.01	-0.01	-	0.01	0.03⋆	0.04	0.06⋆	0.08⋆	0.12†	0.13†	0.26‡
PROMT-NMT	-0.03	-0.02	-0.02	-0.01	-	0.02	0.03	0.05	0.07	0.11⋆	0.12⋆	0.25‡
ONLINE-B	-0.05	-0.04	-0.04	-0.03	-0.02	-	0.01	0.03	0.05	0.09	0.09	0.23‡
HUMAN	-0.06	-0.05	-0.05	-0.04	-0.03	-0.01	-	0.02	0.04	0.08⋆	0.08⋆	0.22‡
ARIEL XV	-0.08	-0.07	-0.07	-0.06	-0.05	-0.03	-0.02	-	0.02	0.06	0.06	0.20‡
AFRL	-0.10	-0.09	-0.09	-0.08	-0.07	-0.05	-0.04	-0.02	-	0.04	0.05	0.18‡
DIDI-NLP	-0.14	-0.13	-0.13	-0.12	-0.11	-0.09	-0.08	-0.06	-0.04	-	0.01	0.14‡
ONLINE-Z	-0.15	-0.14	-0.13	-0.13	-0.12	-0.09	-0.08	-0.06	-0.05	-0.01	-	0.13‡
ZLABS-NLP	-0.28	-0.27	-0.27	-0.26	-0.25	-0.23	-0.22	-0.20	-0.18	-0.14	-0.13	-
score	0.12	0.11	0.11	0.10	0.10	0.07	0.06	0.04	0.03	-0.02	-0.02	-0.15
rank	1–11	1–11	1–11	1–11	1–11	1–11	1–11	1–11	1–11	1–11	1–11	12

Table 36: Head to head comparison for Russian→English systems

	TOHOKU-AIP-NTT	NIUTRANS	OPPO	NICT-KYOTO	ONLINE-B	ONLINE-A	eTRANSLATION	ZLABS-NLP	ONLINE-G	ONLINE-Z
TOHOKU-AIP-NTT	-	0.04	0.10	0.10⋆	0.12‡	0.16‡	0.16‡	0.39‡	0.40‡	0.42‡
NIUTRANS	-0.04	-	0.06	0.06⋆	0.08‡	0.12‡	0.13‡	0.35‡	0.37‡	0.39‡
OPPO	-0.10	-0.06	-	0.00	0.02⋆	0.06⋆	0.07⋆	0.30‡	0.31‡	0.33‡
NICT-KYOTO	-0.10	-0.06	0.00	-	0.02	0.06	0.06	0.29‡	0.30‡	0.32‡
ONLINE-B	-0.12	-0.08	-0.02	-0.02	-	0.04	0.05	0.28‡	0.29‡	0.31‡
ONLINE-A	-0.16	-0.12	-0.06	-0.06	-0.04	-	0.01	0.23‡	0.25‡	0.27‡
eTRANSLATION	-0.16	-0.13	-0.07	-0.06	-0.05	-0.01	-	0.23‡	0.24‡	0.26‡
ZLABS-NLP	-0.39	-0.35	-0.30	-0.29	-0.28	-0.23	-0.23	-	0.01	0.03
ONLINE-G	-0.40	-0.37	-0.31	-0.30	-0.29	-0.25	-0.24	-0.01	-	0.02
ONLINE-Z	-0.42	-0.39	-0.33	-0.32	-0.31	-0.27	-0.26	-0.03	-0.02	-
score	0.18	0.15	0.09	0.08	0.07	0.03	0.02	-0.21	-0.22	-0.24
rank	1–7	1–7	1–7	1–7	1–7	1–7	1–7	8–10	8–10	8–10

Table 37: Head to head comparison for Japanese→English systems

	SRPOL	ONLINE-G	NICT-RUI	ONLINE-B	SJTU-NICT	ONLINE-A	OPPO	ONLINE-Z	CUNI-TRANSFORMER	NICT-KYOTO	VOLCTRANS	PROMT-NMT	TILDE	ZLABS-NLP
SRPOL	-	0.03	0.04	0.04⋆	0.05	0.05⋆	0.08†	0.12†	0.13‡	0.17‡	0.17‡	0.18‡	0.20‡	0.26‡
ONLINE-G	-0.03	-	0.00	0.00	0.01	0.01	0.05	0.09⋆	0.10⋆	0.13†	0.14‡	0.14‡	0.17‡	0.23‡
NICT-RUI	-0.04	0.00	-	0.00	0.01	0.01	0.05⋆	0.09†	0.10†	0.13‡	0.14‡	0.14‡	0.17‡	0.23‡
ONLINE-B	-0.04	0.00	0.00	-	0.01	0.01	0.04	0.09	0.10	0.13⋆	0.14†	0.14†	0.17‡	0.22‡
SJTU-NICT	-0.05	-0.01	-0.01	-0.01	-	0.00	0.04	0.08⋆	0.09†	0.12†	0.13‡	0.13‡	0.16‡	0.22‡
ONLINE-A	-0.05	-0.01	-0.01	-0.01	0.00	-	0.03	0.08	0.09⋆	0.12⋆	0.12†	0.13†	0.15‡	0.21‡
OPPO	-0.08	-0.05	-0.05	-0.04	-0.04	-0.03	-	0.04	0.05	0.09	0.09⋆	0.10⋆	0.12†	0.18‡
ONLINE-Z	-0.12	-0.09	-0.09	-0.09	-0.08	-0.08	-0.04	-	0.01	0.04	0.05	0.05	0.08†	0.14†
CUNI-TRANSFORMER	-0.13	-0.10	-0.10	-0.10	-0.09	-0.09	-0.05	-0.01	-	0.03	0.04	0.04	0.07⋆	0.13†
NICT-KYOTO	-0.17	-0.13	-0.13	-0.13	-0.12	-0.12	-0.09	-0.04	-0.03	-	0.00	0.01	0.03	0.09⋆
VOLCTRANS	-0.17	-0.14	-0.14	-0.14	-0.13	-0.12	-0.09	-0.05	-0.04	0.00	-	0.01	0.03	0.09
PROMT-NMT	-0.18	-0.14	-0.14	-0.14	-0.13	-0.13	-0.10	-0.05	-0.04	-0.01	-0.01	-	0.02	0.08
TILDE	-0.20	-0.17	-0.17	-0.17	-0.16	-0.15	-0.12	-0.08	-0.07	-0.03	-0.03	-0.02	-	0.06
ZLABS-NLP	-0.26	-0.23	-0.23	-0.22	-0.22	-0.21	-0.18	-0.14	-0.13	-0.09	-0.09	-0.08	-0.06	-
score	0.13	0.10	0.10	0.09	0.09	0.08	0.05	0.01	-0.00	-0.04	-0.04	-0.05	-0.07	-0.13
rank	1–14	1–14	1–14	1–14	1–14	1–14	1–14	1–14	1–14	1–14	1–14	1–14	1–14	1–14

Table 38: Head to head comparison for Polish→English systems

	GTCOM	OPPO	ONLINE-B	FACEBOOK-AI	NIUTRANS	VOLCTRANS	ONLINE-Z	ZLABS-NLP	MICROSOFT-STC-INDIA	UEDIN	ONLINE-A	DCU	ONLINE-G	TALP-UPC
GTCOM	-	0.00	0.03	0.03	0.05	0.09	0.20‡	0.20‡	0.22‡	0.22‡	0.27‡	0.28‡	0.60‡	0.65‡
OPPO	0.00	-	0.03	0.03	0.05	0.09	0.20‡	0.20‡	0.22‡	0.22‡	0.27‡	0.28‡	0.60‡	0.65‡
ONLINE-B	-0.03	-0.03	-	0.00	0.03	0.06	0.17‡	0.17‡	0.19‡	0.20†	0.24‡	0.25‡	0.57‡	0.63‡
FACEBOOK-AI	-0.03	-0.03	0.00	-	0.02	0.06⋆	0.17‡	0.17‡	0.19‡	0.19‡	0.24‡	0.25‡	0.57‡	0.62‡
NIUTRANS	-0.05	-0.05	-0.03	-0.02	-	0.03	0.14‡	0.15†	0.17‡	0.17‡	0.22‡	0.23‡	0.55‡	0.60‡
VOLCTRANS	-0.09	-0.09	-0.06	-0.06	-0.03	-	0.11†	0.12†	0.13‡	0.14†	0.18‡	0.19‡	0.51‡	0.57‡
ONLINE-Z	-0.20	-0.20	-0.17	-0.17	-0.14	-0.11	-	0.01	0.02	0.03	0.07	0.08	0.41‡	0.46‡
ZLABS-NLP	-0.20	-0.20	-0.17	-0.17	-0.15	-0.12	-0.01	-	0.02	0.02	0.07	0.08	0.40‡	0.45‡
MICROSOFT-STC-INDIA	-0.22	-0.22	-0.19	-0.19	-0.17	-0.13	-0.02	-0.02	-	0.00	0.05	0.06	0.38‡	0.43‡
UEDIN	-0.22	-0.22	-0.20	-0.19	-0.17	-0.14	-0.03	-0.02	0.00	-	0.05	0.06	0.38‡	0.43‡
ONLINE-A	-0.27	-0.27	-0.24	-0.24	-0.22	-0.18	-0.07	-0.07	-0.05	-0.05	-	0.01	0.33‡	0.38‡
DCU	-0.28	-0.28	-0.25	-0.25	-0.23	-0.19	-0.08	-0.08	-0.06	-0.06	-0.01	-	0.32‡	0.37‡
ONLINE-G	-0.60	-0.60	-0.57	-0.57	-0.55	-0.51	-0.41	-0.40	-0.38	-0.38	-0.33	-0.32	-	0.05⋆
TALP-UPC	-0.65	-0.65	-0.63	-0.62	-0.60	-0.57	-0.46	-0.45	-0.43	-0.43	-0.38	-0.37	-0.05	-
score	0.20	0.20	0.18	0.17	0.15	0.12	0.01	0.00	-0.02	-0.02	-0.07	-0.08	-0.40	-0.45
rank	1–6	1–6	1–6	1–6	1–6	1–6	7–12	7–12	7–12	7–12	7–12	7–12	13	14

Table 39: Head to head comparison for Tamil→English systems

	NiuTrans	Facebook-AI	CUNI-Transfer	Groningen	SRPOL	Helsinki	NRC	UEDIN	UQAM-TanLe	NICT-Kyoto	OPPO
NiuTrans	-	0.00	0.07⋆	0.07⋆	0.10†	0.10⋆	0.11†	0.11‡	0.12†	0.16‡	0.20‡
Facebook-AI	0.00	-	0.07⋆	0.07⋆	0.10⋆	0.10⋆	0.11†	0.11‡	0.12†	0.16‡	0.20‡
CUNI-Transfer	-0.07	-0.07	-	0.00	0.03	0.03	0.04	0.05	0.05	0.09⋆	0.13‡
Groningen	-0.07	-0.07	0.00	-	0.02	0.03	0.04	0.04	0.05	0.09⋆	0.13†
SRPOL	-0.10	-0.10	-0.03	-0.02	-	0.01	0.02	0.02	0.02	0.07⋆	0.11†
Helsinki	-0.10	-0.10	-0.03	-0.03	-0.01	-	0.01	0.01	0.02	0.06⋆	0.10†
NRC	-0.11	-0.11	-0.04	-0.04	-0.02	-0.01	-	0.00	0.01	0.05	0.09⋆
UEDIN	-0.11	-0.11	-0.05	-0.04	-0.02	-0.01	0.00	-	0.01	0.05	0.09⋆
UQAM-TanLe	-0.12	-0.12	-0.05	-0.05	-0.02	-0.02	-0.01	-0.01	-	0.04	0.08⋆
NICT-Kyoto	-0.16	-0.16	-0.09	-0.09	-0.07	-0.06	-0.05	-0.05	-0.04	-	0.04
OPPO	-0.20	-0.20	-0.13	-0.13	-0.11	-0.10	-0.09	-0.09	-0.08	-0.04	-
score	0.17	0.17	0.10	0.10	0.07	0.07	0.06	0.05	0.05	0.01	-0.04
rank	1–2	1–2	3–11	3–11	3–11	3–11	3–11	3–11	3–11	3–11	3–11

Table 40: Head to head comparison for Inuktitut→English systems

	Online-B	GTCOM	Huawei-TSC	VolcTrans	OPPO	Online-Z
Online-B	-	0.01	0.05⋆	0.14‡	0.20‡	0.23‡
GTCOM	-0.01	-	0.04	0.13‡	0.19‡	0.22‡
Huawei-TSC	-0.05	-0.04	-	0.09⋆	0.15‡	0.18‡
VolcTrans	-0.14	-0.13	-0.09	-	0.06⋆	0.09†
OPPO	-0.20	-0.19	-0.15	-0.06	-	0.03
Online-Z	-0.23	-0.22	-0.18	-0.09	-0.03	-
score	0.03	0.02	-0.02	-0.11	-0.16	-0.20
rank	1–3	1–3	1–3	4	5–6	5–6

Table 41: Head to head comparison for Pashto→English systems

	Online-B	GTCOM	Huawei-TSC	VolcTrans	OPPO	Online-Z	Online-G
Online-B	-	0.02	0.03	0.22‡	0.38‡	0.39‡	0.45‡
GTCOM	-0.02	-	0.01	0.19‡	0.36‡	0.37‡	0.43‡
Huawei-TSC	-0.03	-0.01	-	0.18‡	0.35‡	0.36‡	0.42‡
VolcTrans	-0.22	-0.19	-0.18	-	0.16‡	0.18‡	0.23‡
OPPO	-0.38	-0.36	-0.35	-0.16	-	0.01	0.07
Online-Z	-0.39	-0.37	-0.36	-0.18	-0.01	-	0.06
Online-G	-0.45	-0.43	-0.42	-0.23	-0.07	-0.06	-
score	0.17	0.15	0.14	-0.05	-0.21	-0.22	-0.28
rank	1–3	1–3	1–3	4	5–7	5–7	5–7

Table 42: Head to head comparison for Khmer→English systems

	HUMAN-B	HUMAN-A	OPPO	TENCENT-TRANSLATION	HUAWEI-TSC	NIUTRANS	SJTU-NICT	VOLCTRANS	ONLINE-B	ONLINE-A	DONG-NMT	ONLINE-Z	ONLINE-G	ZLABS-NLP
HUMAN-B	-	0.04‡	0.12‡	0.15‡	0.15‡	0.16‡	0.18‡	0.19‡	0.29‡	0.33‡	0.43‡	0.43‡	0.45‡	0.49‡
HUMAN-A	-0.04	-	0.08‡	0.11‡	0.11‡	0.13‡	0.14‡	0.16‡	0.25‡	0.29‡	0.39‡	0.39‡	0.41‡	0.45‡
OPPO	-0.12	-0.08	-	0.03‡	0.03‡	0.04‡	0.06‡	0.07‡	0.16‡	0.21‡	0.31‡	0.31‡	0.32‡	0.36‡
TENCENT-TRANSLATION	-0.15	-0.11	-0.03	-	0.01	0.02	0.03⋆	0.05	0.14‡	0.18‡	0.28‡	0.29‡	0.30‡	0.34‡
HUAWEI-TSC	-0.15	-0.11	-0.03	-0.01	-	0.01	0.03⋆	0.04	0.13‡	0.17‡	0.28‡	0.28‡	0.29‡	0.33‡
NIUTRANS	-0.16	-0.13	-0.04	-0.02	-0.01	-	0.02	0.03	0.12‡	0.16‡	0.27‡	0.27‡	0.28‡	0.32‡
SJTU-NICT	-0.18	-0.14	-0.06	-0.03	-0.03	-0.02	-	0.01	0.10‡	0.15‡	0.25‡	0.25‡	0.27‡	0.30‡
VOLCTRANS	-0.19	-0.16	-0.07	-0.05	-0.04	-0.03	-0.01	-	0.09‡	0.13‡	0.24‡	0.24‡	0.25‡	0.29‡
ONLINE-B	-0.29	-0.25	-0.16	-0.14	-0.13	-0.12	-0.10	-0.09	-	0.04‡	0.15‡	0.15‡	0.16‡	0.20‡
ONLINE-A	-0.33	-0.29	-0.21	-0.18	-0.17	-0.16	-0.15	-0.13	-0.04	-	0.11‡	0.11‡	0.12‡	0.16‡
DONG-NMT	-0.43	-0.39	-0.31	-0.28	-0.28	-0.27	-0.25	-0.24	-0.15	-0.11	-	0.00	0.01	0.05†
ONLINE-Z	-0.43	-0.39	-0.31	-0.29	-0.28	-0.27	-0.25	-0.24	-0.15	-0.11	0.00	-	0.01	0.05⋆
ONLINE-G	-0.45	-0.41	-0.32	-0.30	-0.29	-0.28	-0.27	-0.25	-0.16	-0.12	-0.01	-0.01	-	0.04⋆
ZLABS-NLP	-0.49	-0.45	-0.36	-0.34	-0.33	-0.32	-0.30	-0.29	-0.20	-0.16	-0.05	-0.05	-0.04	-
score	0.57	0.53	0.45	0.42	0.41	0.40	0.39	0.37	0.28	0.24	0.14	0.14	0.12	0.08
rank	1	2	3	4–8	4–8	4–8	4–8	4–8	9	10	11–13	11–13	11–13	14

Table 43: Head to head comparison for English→Chinese systems

	HUMAN	CUNI-DOCTRANSFORMER	OPPO	SRPOL	CUNI-T2T-2018	ETRANSLATION	CUNI-TRANSFORMER	UEDIN-CUNI	ONLINE-B	ONLINE-Z	ONLINE-A	ONLINE-G	ZLABS-NLP
HUMAN	-	0.11‡	0.12‡	0.15‡	0.20‡	0.21‡	0.22‡	0.33‡	0.61‡	0.64‡	0.65‡	0.87‡	1.41‡
CUNI-DOCTRANSFORMER	-0.11	-	0.01	0.04†	0.09‡	0.11‡	0.11‡	0.22‡	0.50‡	0.53‡	0.54‡	0.76‡	1.31‡
OPPO	-0.12	-0.01	-	0.03⋆	0.08‡	0.10‡	0.10‡	0.22‡	0.49‡	0.52‡	0.53‡	0.75‡	1.30‡
SRPOL	-0.15	-0.04	-0.03	-	0.05⋆	0.06‡	0.07†	0.18‡	0.46‡	0.49‡	0.50‡	0.72‡	1.26‡
CUNI-T2T-2018	-0.20	-0.09	-0.08	-0.05	-	0.02	0.02	0.14‡	0.41‡	0.44‡	0.45‡	0.67‡	1.22‡
ETRANSLATION	-0.21	-0.11	-0.10	-0.06	-0.02	-	0.01	0.12‡	0.39‡	0.42‡	0.43‡	0.66 ‡	1.20‡
CUNI-TRANSFORMER	-0.22	-0.11	-0.10	-0.07	-0.02	-0.01	-	0.11‡	0.39‡	0.42‡	0.43‡	0.65‡	1.19‡
UEDIN-CUNI	-0.33	-0.22	-0.22	-0.18	-0.14	-0.12	-0.11	-	0.27‡	0.30‡	0.31‡	0.54‡	1.08‡
ONLINE-B	-0.61	-0.50	-0.49	-0.46	-0.41	-0.39	-0.39	-0.27	-	0.03	0.04	0.26‡	0.81‡
ONLINE-Z	-0.64	-0.53	-0.52	-0.49	-0.44	-0.42	-0.42	-0.30	-0.03	-	0.01	0.23‡	0.78‡
ONLINE-A	-0.65	-0.54	-0.53	-0.50	-0.45	-0.43	-0.43	-0.31	-0.04	-0.01	-	0.22‡	0.77‡
ONLINE-G	-0.87	-0.76	-0.75	-0.72	-0.67	-0.66	-0.65	-0.54	-0.26	-0.23	-0.22	-	0.54‡
ZLABS-NLP	-1.41	-1.31	-1.30	-1.26	-1.22	-1.20	-1.19	-1.08	-0.81	-0.78	-0.77	-0.54	-
score	0.65	0.55	0.54	0.51	0.46	0.44	0.43	0.32	0.05	0.02	0.01	-0.22	-0.76
rank	1	2–3	2–3	4	5–7	5–7	5–7	8	9–11	9–11	9–11	12	13

Table 44: Head to head comparison for English→Czech systems

	HUMAN-B	OPPO	TOHOKU-AIP-NTT	HUMAN-A	ONLINE-B	TENCENT-TRANSLATION	VOLCTRANS	ONLINE-A	ETRANSLATION	HUMAN-C	AFRL	UEDIN	PROMT-NMT	ONLINE-Z	ONLINE-G	ZLABS-NLP	WMTBIOMEDBASELINE
HUMAN-B	-	0.07⋆	0.10‡	0.12⋆	0.15‡	0.18‡	0.24‡	0.25‡	0.26‡	0.27‡	0.31‡	0.32‡	0.32‡	0.44‡	0.69‡	0.85‡	0.91‡
OPPO	-0.07	-	0.03	0.05	0.08⋆	0.11‡	0.17‡	0.17‡	0.18‡	0.20‡	0.24‡	0.24‡	0.25‡	0.37‡	0.61‡	0.77‡	0.83‡
TOHOKU-AIP-NTT	-0.10	-0.03	-	0.02	0.05	0.08⋆	0.14‡	0.15‡	0.16‡	0.17‡	0.21‡	0.22‡	0.22‡	0.34‡	0.59‡	0.75‡	0.81‡
HUMAN-A	-0.12	-0.05	-0.02	-	0.03⋆	0.06†	0.12‡	0.12‡	0.13‡	0.15‡	0.19‡	0.19‡	0.20‡	0.32‡	0.57‡	0.72‡	0.78‡
ONLINE-B	-0.15	-0.08	-0.05	-0.03	-	0.03	0.09‡	0.09‡	0.10‡	0.12‡	0.16‡	0.17‡	0.17‡	0.29‡	0.54‡	0.69‡	0.75‡
TENCENT-TRANSLATION	-0.18	-0.11	-0.08	-0.06	-0.03	-	0.06‡	0.06‡	0.07†	0.09‡	0.12‡	0.13‡	0.14‡	0.26‡	0.50‡	0.66‡	0.72‡
VOLCTRANS	-0.24	-0.17	-0.14	-0.12	-0.09	-0.06	-	0.00	0.01	0.03	0.07	0.08†	0.08†	0.20‡	0.45‡	0.60‡	0.66‡
ONLINE-A	-0.25	-0.17	-0.15	-0.12	-0.09	-0.06	0.00	-	0.01	0.02	0.06	0.07†	0.08†	0.20‡	0.44‡	0.60‡	0.66‡
ETRANSLATION	-0.26	-0.18	-0.16	-0.13	-0.10	-0.07	-0.01	-0.01	-	0.01	0.05	0.06†	0.06‡	0.19‡	0.43‡	0.59‡	0.65‡
HUMAN-C	-0.27	-0.20	-0.17	-0.15	-0.12	-0.09	-0.03	-0.02	-0.01	-	0.04	0.05⋆	0.05⋆	0.17‡	0.42‡	0.58‡	0.64‡
AFRL	-0.31	-0.24	-0.21	-0.19	-0.16	-0.12	-0.07	-0.06	-0.05	-0.04	-	0.01	0.01⋆	0.13‡	0.38‡	0.54‡	0.60‡
UEDIN	-0.32	-0.24	-0.22	-0.19	-0.17	-0.13	-0.08	-0.07	-0.06	-0.05	-0.01	-	0.00	0.13†	0.37‡	0.53‡	0.59‡
PROMT-NMT	-0.32	-0.25	-0.22	-0.20	-0.17	-0.14	-0.08	-0.08	-0.06	-0.05	-0.01	0.00	-	0.12‡	0.37‡	0.53‡	0.59‡
ONLINE-Z	-0.44	-0.37	-0.34	-0.32	-0.29	-0.26	-0.20	-0.20	-0.19	-0.17	-0.13	-0.13	-0.12	-	0.25‡	0.40‡	0.46‡
ONLINE-G	-0.69	-0.61	-0.59	-0.57	-0.54	-0.50	-0.45	-0.44	-0.43	-0.42	-0.38	-0.37	-0.37	-0.25	-	0.16	0.22†
ZLABS-NLP	-0.85	-0.77	-0.75	-0.72	-0.69	-0.66	-0.60	-0.60	-0.59	-0.58	-0.54	-0.53	-0.53	-0.40	-0.16	-	0.06
WMTBIOMEDBASELINE	-0.91	-0.83	-0.81	-0.78	-0.75	-0.72	-0.66	-0.66	-0.65	-0.64	-0.60	-0.59	-0.59	-0.46	-0.22	-0.06	-
score	0.57	0.49	0.47	0.45	0.42	0.39	0.33	0.32	0.31	0.30	0.26	0.25	0.25	0.13	-0.12	-0.28	-0.34
rank	1	2–6	2–6	2–6	2–6	2–6	7–13	7–13	7–13	7–13	7–13	7–13	7–13	14	15–17	15–17	15–17

Table 45: Head to head comparison for English→German systems

	HUMAN	MULTILINGUAL-UBIQUS	CUNI-TRANSFER	NRC	FACEBOOK-AI	NICT_KYOTO	GRONINGEN	HELSINKI	SRPOL	UQAM-TANLE	UEDIN	OPPO
HUMAN	-	0.15	0.17†	0.21‡	0.21‡	0.21†	0.24‡	0.28‡	0.29‡	0.49‡	0.49‡	0.96‡
MULTILINGUAL-UBIQUS	-0.15	-	0.02⋆	0.06‡	0.06‡	0.06⋆	0.09‡	0.13‡	0.14‡	0.34‡	0.34‡	0.81‡
CUNI-TRANSFER	-0.17	-0.02	-	0.04†	0.04	0.04	0.07†	0.11‡	0.13‡	0.32‡	0.33‡	0.79‡
NRC	-0.21	-0.06	-0.04	-	0.00	0.01	0.03	0.07	0.09	0.29‡	0.29‡	0.75‡
FACEBOOK-AI	-0.21	-0.06	-0.04	0.00	-	0.00	0.03	0.07†	0.09†	0.28‡	0.29‡	0.75‡
NICT-KYOTO	-0.21	-0.06	-0.04	-0.01†	0.00	-	0.02†	0.07‡	0.08‡	0.28‡	0.28‡	0.75‡
GRONINGEN	-0.24	-0.09	-0.07	-0.03	-0.03	-0.02	-	0.04	0.06	0.26‡	0.26‡	0.72‡
HELSINKI	-0.28	-0.13	-0.11	-0.07	-0.07	-0.07	-0.04	-	0.01	0.21‡	0.21‡	0.68‡
SRPOL	-0.29	-0.14	-0.13	-0.09	-0.09	-0.08	-0.06	-0.01	-	0.20‡	0.20‡	0.67‡
UQAM-TANLE	-0.49	-0.34	-0.32	-0.29	-0.28	-0.28	-0.26	-0.21	-0.20	-	0.00	0.47‡
UEDIN	-0.49	-0.34	-0.33	-0.29	-0.29	-0.28	-0.26	-0.21	-0.20	0.00	-	0.47‡
OPPO	-0.96	-0.81	-0.79	-0.75	-0.75	-0.75	-0.72	-0.68	-0.67	-0.47	-0.47	-
score	0.57	0.42	0.41	0.37	0.37	0.36	0.34	0.30	0.28	0.08	0.08	-0.38
rank	1–2	1–2	3–9	3–9	3–9	3–9	3–9	3–9	3–9	10–11	10–11	12

Table 46: Head to head comparison for English→Inuktitut systems

	HUMAN	NIUTRANS	TOHOKU-AIP-NTT	OPPO	ENMT	NICT-KYOTO	ONLINE-A	ONLINE-B	ZLABS-NLP	ONLINE-Z	SJTU-NICT	ONLINE-G
HUMAN	-	0.07†	0.08†	0.08†	0.08†	0.20‡	0.23‡	0.24‡	0.42‡	0.54‡	0.71‡	0.74‡
NIUTRANS	-0.07	-	0.01	0.01	0.01	0.13‡	0.15‡	0.17‡	0.34‡	0.47‡	0.63‡	0.67‡
TOHOKU-AIP-NTT	-0.08	-0.01	-	0.00	0.00	0.12‡	0.15‡	0.16‡	0.34‡	0.46‡	0.63‡	0.66‡
OPPO	-0.08	-0.01	0.00	-	0.00	0.12‡	0.15‡	0.16‡	0.34‡	0.46‡	0.63‡	0.66‡
ENMT	-0.08	-0.01	0.00	0.00	-	0.12‡	0.14‡	0.16‡	0.33‡	0.46‡	0.62‡	0.66‡
NICT-KYOTO	-0.20	-0.13	-0.12	-0.12	-0.12	-	0.03	0.04⋆	0.22‡	0.34‡	0.51‡	0.54‡
ONLINE-A	-0.23	-0.15	-0.15	-0.15	-0.14	-0.03	-	0.01	0.19‡	0.32‡	0.48‡	0.51‡
ONLINE-B	-0.24	-0.17	-0.16	-0.16	-0.16	-0.04	-0.01	-	0.18‡	0.30‡	0.47‡	0.50‡
ZLABS-NLP	-0.42	-0.34	-0.34	-0.34	-0.33	-0.22	-0.19	-0.18	-	0.13‡	0.29‡	0.32‡
ONLINE-Z	-0.54	-0.47	-0.46	-0.46	-0.46	-0.34	-0.32	-0.30	-0.13	-	0.16‡	0.20‡
SJTU-NICT	-0.71	-0.63	-0.63	-0.63	-0.62	-0.51	-0.48	-0.47	-0.29	-0.16	-	0.03
ONLINE-G	-0.74	-0.67	-0.66	-0.66	-0.66	-0.54	-0.51	-0.50	-0.32	-0.20	-0.03	-
score	0.58	0.50	0.50	0.50	0.49	0.38	0.35	0.34	0.16	0.03	-0.13	-0.16
rank	1	2–5	2–5	2–5	2–5	6–8	6–8	6–8	9	10	11–12	11–12

Table 47: Head to head comparison for English→Japanese systems

	HUMAN	SRPOL	ETRANSLATION	VOLCTRANS	TILDE	ONLINE-G	OPPO	NICT-KYOTO	TILDE	CUNI-TRANSFORMER	ONLINE-B	SJTU-NICT	ONLINE-A	ONLINE-Z	ZLABS-NLP
HUMAN	-	0.18‡	0.24‡	0.29‡	0.32‡	0.36‡	0.36‡	0.37‡	0.40‡	0.42‡	0.44‡	0.45‡	0.58‡	0.73‡	1.21‡
SRPOL	-0.18	-	0.06⋆	0.11†	0.14‡	0.18‡	0.18‡	0.19‡	0.22‡	0.24‡	0.26‡	0.27‡	0.40‡	0.55‡	1.03‡
ETRANSLATION	-0.24	-0.06	-	0.05	0.09†	0.12†	0.12‡	0.14†	0.16‡	0.18‡	0.20‡	0.22‡	0.34‡	0.49‡	0.97‡
VOLCTRANS	-0.29	-0.11	-0.05	-	0.03	0.07	0.07‡	0.08	0.11‡	0.13†	0.15‡	0.16‡	0.29‡	0.44‡	0.92‡
TILDE	-0.32	-0.14	-0.09	-0.03	-	0.03	0.04⋆	0.05	0.08⋆	0.09	0.11‡	0.13‡	0.25‡	0.41‡	0.89‡
ONLINE-G	-0.36	-0.18	-0.12	-0.07	-0.03	-	0.01	0.02	0.04	0.06	0.08†	0.10†	0.22‡	0.38‡	0.85‡
OPPO	-0.36	-0.18	-0.12	-0.07	-0.04	-0.01	-	0.01	0.04	0.06	0.07⋆	0.09⋆	0.21‡	0.37‡	0.85‡
NICT-KYOTO	-0.37	-0.19	-0.14	-0.08	-0.05	-0.02	-0.01⋆	-	0.03⋆	0.04	0.06‡	0.08‡	0.20‡	0.36‡	0.84‡
TILDE	-0.40	-0.22	-0.16	-0.11	-0.08	-0.04	-0.04	-0.03	-	0.02	0.04	0.05	0.18‡	0.33‡	0.81‡
CUNI-TRANSFORMER	-0.42	-0.24	-0.18	-0.13	-0.09	-0.06	-0.06	-0.04	-0.02	-	0.02⋆	0.04⋆	0.16‡	0.31‡	0.79‡
ONLINE-B	-0.44	-0.26	-0.20	-0.15	-0.11	-0.08	-0.07	-0.06	-0.04	-0.02	-	0.02	0.14‡	0.30‡	0.77‡
SJTU-NICT	-0.45	-0.27	-0.22	-0.16	-0.13	-0.10	-0.09	-0.08	-0.05	-0.04	-0.02	-	0.12†	0.28‡	0.76‡
ONLINE-A	-0.58	-0.40	-0.34	-0.29	-0.25	-0.22	-0.21	-0.20	-0.18	-0.16	-0.14	-0.12	-	0.16‡	0.63‡
ONLINE-Z	-0.73	-0.55	-0.49	-0.44	-0.41	-0.38	-0.37	-0.36	-0.33	-0.31	-0.30	-0.28	-0.16	-	0.48‡
ZLABS-NLP	-1.21	-1.03	-0.97	-0.92	-0.89	-0.85	-0.85	-0.84	-0.81	-0.79	-0.77	-0.76	-0.63	-0.48	-
score	0.67	0.49	0.43	0.38	0.35	0.32	0.31	0.30	0.27	0.26	0.24	0.22	0.10	-0.06	-0.54
rank	1	2	3–8	3–8	3–8	3–8	3–8	3–8	9–10	9–10	11–12	11–12	13	14	15

Table 48: Head to head comparison for English→Polish systems

	HUMAN	ONLINE-G	OPPO	ARIEL XV	ONLINE-B	PROMT-NMT	DIDI-NLP	ONLINE-A	ZLABS-NLP	ONLINE-Z
HUMAN	-	0.21‡	0.22‡	0.28‡	0.35‡	0.43‡	0.46‡	0.60‡	0.65‡	0.67‡
ONLINE-G	-0.21	-	0.01	0.06⋆	0.13‡	0.22‡	0.25‡	0.39‡	0.43‡	0.46‡
OPPO	-0.22	-0.01	-	0.06	0.13‡	0.21‡	0.24‡	0.38‡	0.43‡	0.45‡
ARIEL XV	-0.28	-0.06	-0.06	-	0.07†	0.15‡	0.18‡	0.32‡	0.37‡	0.39‡
ONLINE-B	-0.35	-0.13	-0.13	-0.07	-	0.08†	0.11‡	0.25‡	0.30‡	0.32‡
PROMT-NMT	-0.43	-0.22	-0.21	-0.15	-0.08	-	0.03⋆	0.17‡	0.22‡	0.24‡
DIDI-NLP	-0.46	-0.25	-0.24	-0.18	-0.11	-0.03	-	0.14‡	0.19‡	0.21‡
ONLINE-A	-0.60	-0.39	-0.38	-0.32	-0.25	-0.17	-0.14	-	0.05	0.07†
ZLABS-NLP	-0.65	-0.43	-0.43	-0.37	-0.30	-0.22	-0.19	-0.05	-	0.02
ONLINE-Z	-0.67	-0.46	-0.45	-0.39	-0.32	-0.24	-0.21	-0.07	-0.02	-
score	0.68	0.47	0.46	0.40	0.34	0.25	0.22	0.08	0.04	0.01
rank	1	2–4	2–4	2–4	5	6	7	8–10	8–10	8–10

Table 49: Head to head comparison for English→Russian systems

	HUMAN	FACEBOOK-AI	GTCOM	ONLINE-B	OPPO	ONLINE-A	VOLCTRANS	ONLINE-Z	ZLABS-NLP	MICROSOFT-STC-INDIA	UEDIN	GRONINGEN	DCU	TALP-UPC	ONLINE-G	SJTU-NICT
HUMAN	-	0.10‡	0.25‡	0.27‡	0.28‡	0.31‡	0.34‡	0.44‡	0.45‡	0.47‡	0.53‡	0.61‡	0.77‡	1.17‡	1.48‡	1.58‡
FACEBOOK-AI	-0.10	-	0.15‡	0.17‡	0.18‡	0.21‡	0.24‡	0.34‡	0.36‡	0.37‡	0.43‡	0.51‡	0.67‡	1.07‡	1.38‡	1.48‡
GTCOM	-0.25	-0.15	-	0.02	0.03	0.06	0.09	0.19‡	0.21‡	0.22‡	0.28‡	0.36‡	0.52‡	0.92‡	1.23‡	1.33‡
ONLINE-B	-0.27	-0.17	-0.02	-	0.01	0.03	0.07	0.17‡	0.18‡	0.19‡	0.26‡	0.34‡	0.50‡	0.90‡	1.21‡	1.31‡
OPPO	-0.28	-0.18	-0.03	-0.01	-	0.02	0.06	0.15‡	0.17‡	0.18‡	0.25‡	0.33‡	0.49‡	0.89‡	1.20‡	1.30‡
ONLINE-A	-0.31	-0.21	-0.06	-0.03	-0.02	-	0.03	0.13‡	0.15†	0.16‡	0.23‡	0.30‡	0.46‡	0.86‡	1.17‡	1.28‡
VOLCTRANS	-0.34	-0.24	-0.09	-0.07	-0.06	-0.03	-	0.10‡	0.12⋆	0.13†	0.19‡	0.27‡	0.43‡	0.83‡	1.14‡	1.24‡
ONLINE-Z	-0.44	-0.34	-0.19	-0.17	-0.15	-0.13	-0.10	-	0.02	0.03	0.09	0.17⋆	0.33‡	0.73‡	1.04‡	1.15‡
ZLABS-NLP	-0.45	-0.36	-0.21	-0.18	-0.17	-0.15	-0.12	-0.02	-	0.01	0.08	0.15†	0.31‡	0.71‡	1.02‡	1.13‡
MICROSOFT-STC-INDIA	-0.47	-0.37	-0.22	-0.19	-0.18	-0.16	-0.13	-0.03	-0.01	-	0.07	0.14⋆	0.30‡	0.70‡	1.01‡	1.12‡
UEDIN	-0.53	-0.43	-0.28	-0.26	-0.25	-0.23	-0.19	-0.09	-0.08	-0.07	-	0.08	0.24‡	0.64‡	0.95‡	1.05‡
GRONINGEN	-0.61	-0.51	-0.36	-0.34	-0.33	-0.30	-0.27	-0.17	-0.15	-0.14	-0.08	-	0.16‡	0.56‡	0.87‡	0.97‡
DCU	-0.77	-0.67	-0.52	-0.50	-0.49	-0.46	-0.43	-0.33	-0.31	-0.30	-0.24	-0.16	-	0.40‡	0.71‡	0.81‡
TALP-UPC	-1.17	-1.07	-0.92	-0.90	-0.89	-0.86	-0.83	-0.73	-0.71	-0.70	-0.64	-0.56	-0.40	-	0.31‡	0.41‡
ONLINE-G	-1.48	-1.38	-1.23	-1.21	-1.20	-1.17	-1.14	-1.04	-1.02	-1.01	-0.95	-0.87	-0.71	-0.31	-	0.10⋆
SJTU-NICT	-1.58	-1.48	-1.33	-1.31	-1.30	-1.28	-1.24	-1.15	-1.13	-1.12	-1.05	-0.97	-0.81	-0.41	-0.10	-
score	0.76	0.66	0.51	0.49	0.48	0.46	0.42	0.33	0.31	0.30	0.23	0.15	-0.01	-0.41	-0.72	-0.82
rank	1	2	3–7	3–7	3–7	3–7	3–7	8–12	8–12	8–12	8–12	8–12	13	14	15	16

Table 50: Head to head comparison for English→Tamil systems

B Translator Brief: Sentence-Split News Test Sets

Translator Brief

In this project we wish to translate online news articles for use in evaluation of Machine Translation (MT). The translations produced by you will be compared against the translations produced by a variety of different MT systems. They will be released to the research community to provide a benchmark, or "gold-standard" measure for translation quality. The translation therefore needs to be a high-quality rendering of the source text into the target language, as if it was news written directly in the target language. However there are some constraints imposed by the intended usage:

- All translations should be **"from scratch", without post-editing from MT**. Using post-editing would bias the evaluation, so we need to avoid it. We can detect post-editing so will reject translations that are post-edited.
- Translation should **preserve the sentence boundaries.** The source texts are provided with exactly one sentence per line, and the translations should be the same, one sentence per line.
- Translators should **avoid inserting parenthetical explanations** into the translated text and obviously **avoid losing any pieces of information** from the source text.

We will check a sample of the translations for quality, and we will check the entire set for evidence of post-editing.

The source files will be delivered as text files (sometimes known as "notepad" files), with one sentence per line. We need the translations to be returned in the same format. If you prefer to receive the text in a different format, then please let us know as we may be able to accommodate it.

C Translator Brief: Paragraph-Split News Test Sets

Translator Brief

In this project we wish to translate online news articles for use in evaluation of Machine Translation (MT). The translations produced by you will be compared against the translations produced by a variety of different MT systems. They will be released to the research community to provide a benchmark, or "gold-standard" measure for translation quality. The translation therefore needs to be a high-quality rendering of the source text into the target language, as if it was news written directly in the target language. However there are some constraints imposed by the intended usage:

- All translations should be **"from scratch", without post-editing from MT**. Using post-editing would bias the evaluation, so we need to avoid it. We can detect post-editing so will reject translations that are post-edited.
- Translation should **preserve paragraph or newline boundaries and blank lines.** The source texts are formatted as short paragraphs separated by blank lines. We need this formatting preserved so that we can align the sources and translations.
- Translators should **avoid inserting parenthetical explanations** into the translated text and obviously **avoid losing any pieces of information** from the source text.

We will check a sample of the translations for quality, and we will check the entire set for evidence of post-editing.

The source files will be delivered as text files (sometimes known as "notepad" files). We need the translations to be returned in the same format, ideally with utf8 encoding. If you prefer to receive the text in a different format, then please let us know as we may be able to accommodate it.

Findings of the First Shared Task on Lifelong Learning Machine Translation

Loïc Barrault
University of Sheffield
l.barrault@sheffield.ac.uk

Magdalena Biesialska & Marta R. Costa-jussà
Universitat Politècnica de Catalunya
first.last@upc.edu

Fethi Bougares
LIUM
fethi.bougares@univ-lemans.fr

Olivier Galibert
LNE
Olivier.Galibert@lne.fr

Abstract

A lifelong learning system can adapt to new data without forgetting previously acquired knowledge. In this paper, we introduce the first benchmark for lifelong learning machine translation. For this purpose, we provide training, lifelong and test data sets for two language pairs: English-German and English-French. Additionally, we report the results of our baseline systems, which we make available to the public. The goal of this shared task is to encourage research on the emerging topic of lifelong learning machine translation.

1 Introduction

Lifelong learning can be defined as the ability to continually acquire new and retain previous knowledge. This ability characterizes humankind, but it is also reflected in several artificial intelligence systems (Parisi et al., 2019; Biesialska et al., 2020). There are many challenges that have to be solved in order to achieve this goal of continual adaptation, among which catastrophic forgetting (French, 1999) seems to be the most relevant.

Lifelong learning is very useful in the area of machine translation (MT), as it allows MT systems to adapt to new vocabularies and topics, and produce accurate translations across time. Currently, there are no previous works that systematically try to solve the problem. This may be due to the lack of a benchmark to address the challenge (Biesialska et al., 2020).

In this context, the main goal of the shared task on lifelong learning for MT is to develop systems that can self-adapt relying solely on domain expert data and are then freed from the necessity of machine learning expertise. What is more, this shared task also allows to investigate several MT research directions, such as: the continuous training/adaptation techniques; the preparation of additional publicly available corpora and evaluation sets; the active learning methods via a controlled simulated environment; the unsupervised adaptation of MT systems; the document-level approaches and the development and evaluation of MT systems across time.

2 Related work and tasks

As mentioned in the introduction, there are not really any works in MT properly evaluating lifelong learning systems. However, there is a long history of studies in related tasks that are useful for addressing the lifelong learning objective.

Domain adaptation is based on the premise that the system can adapt to a target domain known in advance. This has been widely studied earlier for statistical MT e.g. (Koehn and Schroeder, 2007) and, more recently, for neural MT e.g. (Luong and Manning, 2015)).

Instance-based adaptation exploits similarity between training and inference instances (Li et al., 2018), also in unsupervised scenarios (Farajian et al., 2017). These studies have even led to the creation of adaptive MT commercial toolkits (Federico, 2018). Importantly, in this task there is no target domain data available.

Unsupervised learning focuses on using monolingual corpora to train the translation system, without relying on any parallel corpora (Artetxe et al., 2018; Lample et al., 2018).

Active learning aims at selecting the most useful source sentences from a monolingual set and query their translation. This selection needs to minimize the post-edited cost and maximize the improvement of a finetuned model (Liu et al., 2018).

Interactive learning relies on a joint collaboration between a human and an MT system to obtain

Proceedings of the 5th Conference on Machine Translation (WMT), pages 56–64

high-quality translations while reducing the human effort in the process (Peris et al., 2016).

Our lifelong learning setting differs from both domain and instance-based adaptation, as it depends on the target data (called the lifelong data). The lifelong data set, unlike the training data, is unsupervised. Therefore, it is advisable to use techniques such as unsupervised learning, active learning, or interactive learning to approach the task.

3 Overview of the system and environment

The toolchain developed to evaluate the autonomous systems is described in Figure 1. It is made of four parts:

- the input datasets (purple on Figure 1), see section 3.1;
- the four blocks of the system (green on Figure 1 to be modified to include your own system), see appendix A for more details;
- the user simulation (orange on Figure 1), see section A.5;
- the evaluation blocks (blue on Figure 1), see section 3.2

Note that input datasets, user simulation and evaluation blocks are fixed and guarantee the reproducibility of the experiments. Participants are free to edit the four blocks of the system in order to include their own code. Once your code is included in this toolchain, the system will run automatically and the BEAT platform is responsible for managing the data exchanges between the different blocks of the architecture. Thus, you don't need to take care about the communication between blocks, especially, the interaction between the system and the user simulation is automatic.

3.1 Datasets

Two different datasets are available: the **training** data and what we called **lifelong** data. The **training** data is used to train the preprocessing system (eventually) and the initial system in a supervised way. Source text along with the translation of all documents included in this set are available at any time during the lifelong MT process. Note that no development data is provided, meaning that it is up to the participants to decide how to split the training data into train and development (if one is needed).

This year, we used the Europarl and NewsCommentary corpora as training data as they have document information along with their production dates. This represents between 50M and 58.6M words per language depending on the considered language pair (see details in Table 1).

The **lifelong** data is available in a sequential manner: each document is processed one after the other to simulate the process along time. This data is unsupervised, meaning that no reference translation is provided (they correspond to the data to translate every day). The system has to provide translations for those documents that will be evaluated.

We used the WMT14 English to French and English to German corpus as lifelong learning data. While this allows for comparison with systems that participated in WMT14 News translation shared task, one must keep in mind that the training data is much smaller than what was available for the shared task at the time. The aim here is to demonstrate the effectiveness of the continuous adaptation when compared to a baseline system that does not evolve (lower bound) and the best supervised system (retrained with all available data). In the future, we will extend the lifelong learning data to include that from 2014 up to the most recent one.

Training data (from 01.01.1996 to 31.12.2013)				
	English	French	English	German
#Documents	15218		15472	
#Segments	2308516		2246090	
#Words	55.6M	58.6M	53.6M	50.4M
Lifelong data (newstest2014)				
	English	French	English	German
#Documents	176		164	
#Segments	3003		3003	
#Words	62.3k	69.6k	59.3k	55.1k

Table 1: Statistics of the newstest2014 English-French and English-German corpora.

3.2 Evaluation

The evaluation is performed in the *mt_evaluation* and *BLEU_collate* blocks. The first block is aimed at collecting scoring statistics for the document being currently processed. In our case, it will correspond to the BLEU modified n-gram precisions. The second block will aggregate those statistics along with the penalisation in order to provide a final score for the system.

Each time the user simulation is asked for help,

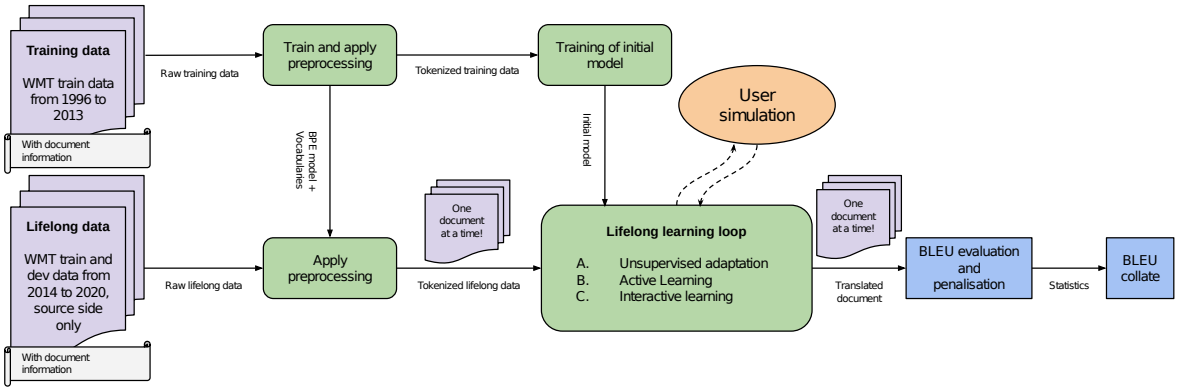


Figure 1: Flowchart of the lifelong learning MT system running on the BEAT platform.

a penalisation is calculated based o the request. The final penalised score S_{pen} corresponds to the following score:

$$S_{pen} = S_{adapt} + (S_{imp} - S_{cor})$$

with S_{adapt} being the score of the adapted system and S_{imp} and S_{cor} are the scores of this system where all sentences requested to the user simulation are considered entirely wrong and correct, respectively. Note that in the case of BLEU, the brevity penalty is not impacted by this calculation, only the correct n-gram counts will be decreased proportionally to the sentence requested for translation. For more details, see (Prokopalo et al., 2020).

4 Baseline systems

Integrating an NMT system in the BEAT platform requires to rethink the code so that everything is done in memory. We chose to use the nmtpytorch toolkit to implement the baseline systems (Caglayan et al., 2017).

Our baseline systems consists of a 2-layer bidirectionnal GRU (Cho et al., 2014) encoder and a 2-layer Conditional GRU decoder (Sennrich et al., 2017) equipped with an attention mechanism (Bahdanau et al., 2014) as implemented in nmtpytorch.

Given a source sequence of embeddings $X=\{x_1, \ldots, x_S\}$ and a target sequence of embeddings $Y=\{y_1, \ldots, y_T\}$, the bidirectional encoder first computes the sequence of annotations corresponding to the concatenation of the hidden states of the two GRU $A=\{a_1, \ldots, a_S\}$. At a given timestep t of decoding, the output layer estimates the probability of the next target word y_t as follows:

$$\begin{aligned} d_t &= \text{GRU}(y_{t-1}, d'_{t-1}) \\ c_t &= \text{Attention}(A, \text{query} \leftarrow d_t) \\ d'_t &= \text{GRU}'(c_t, d_t) \\ o_t &= \tanh(\mathbf{W_c} c_t + \mathbf{W_d} d'_t + \mathbf{W_y} y_{t-1}) \\ l_t &= \mathbf{W_o}(\mathbf{W_b} o_t + b_b) + b_o \\ P(y_t|X, Y_{<t}) &= \text{softmax}(l_t) \end{aligned} \quad (1)$$

For a single training sample, we then maximise the joint likelihood of source and target sentences:

$$\text{L}(X, Y) = \sum_{t=1}^{T} \log\left(P\left(y_t|X, Y_{<t}\right)\right) \quad (2)$$

5 Adaptation techniques

The first adaptation technique used is rather simple. It consists of selecting N sentences from training data that are the closest to the sentences in the document. The chosen similarity metric is the cosine between sentence embeddings obtained by a simple average of word embeddings, as described in (Arora et al., 2017). This data is then used to finetune the initial model for maximum 10 epochs with a learning rate of 0.00004, which is ten times smaller than during initial training of the model.

Furthermore, we employed an active learning strategy as an adaptation method. In principle, there are two steps involved. Firstly, the model provides a translation for each document from the lifelong learning corpus. As the lifelong learning data are unsupervised; therefore, a quality estimation (QE) technique is used to evaluate the quality of the translations without any access to a reference translation. Every document is evaluated

using sentence-level HTER scores (Specia et al., 2018). Secondly, an OpenKiwi QE model (Kepler et al., 2019) is used to rank the sentences according to their quality, and those with the worst HTER score are sent to the user simulation (active learning), which provides the correct translation of the selected sentences. This process implies the penalisation of the BLEU score as explained in Section 3.

6 Experimental setup

The dimensions of embeddings and GRU hidden states are set to 128 and 256, respectively. The embeddings are shared in the decoder (Press and Wolf, 2017). We use ADAM (Kingma and Ba, 2014) as the optimiser and set the learning rate and mini-batch size to 0.0004 and 64, respectively. Regularisation is done by means of a weight decay of $1e{-}5$ and the use of dropout on the embeddings, the source context and the output (set at 0.4) (Srivastava et al., 2014). We clip the gradients if the norm of the full parameter vector exceeds 1 (Pascanu et al., 2013).

The data is processed by a joint BPE model with 30k merge operations (Sennrich et al., 2016a). This leads to respectively 20.7k and 25.1k units for English and French and 17.2k and 26.5k units for English and German, respectively.

We train each model for a maximum of 100 epochs and early stop the training if validation BLEU (Papineni et al., 2002) does not improve for 10 epochs (Figure 2). We also halve the learning rate if no improvement is obtained for three epochs. The number of learnable parameters is around 8.7M for En-Fr and 8.5M for En-De.

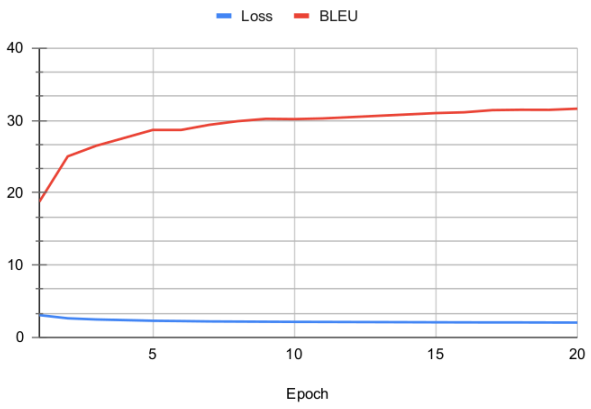


Figure 2: Training loss and BLEU scores for the English→German MT system.

7 Results

The results of the baseline systems and the adapted ones can be found in Table 2.

	English→French	English→German
Baseline		
SHEFFIELD	25.7	15.6
UPC	26.2	14.7
Data selection + finetuning		
SHEFFIELD	26.4	15.5
UPC	26.4	15.1

Table 2: BLEU scores on the newstest2014 English→French and English→German.

Results show that a simple data selection method along with finetuning can provide a small improvement of the system's performance for English to French. German is known to be a more complicated language, as demonstrated by the lower results and the inefficient effect of the adaptation method.

8 Discussion and next year evaluation

We can see that the task, given the very constrained data is very hard. A simple comparison with the results of the systems that participated in the WMT14 News translation task shows more than 10 BLEU points difference. We insist on the fact that the main goal of the challenge is to provide new methods to incrementally adapt the model to incoming documents. Without loss of generality, it is very probable that even with a better baseline system (trained on more data), the adapted models would exhibit a similar improvement.

Many questions and challenges remain open as to how lifelong learning for MT should be implemented. Next year, we ought to push further the evaluation by improving the QE model in order to better select the sentences to be sent to the user Simulation (Active Learning module). Hence, this will require to reconsider how the systems are evaluated. This year, we introduced a way of penalising the systems but without corresponding results.

We hope to have more participants bringing new ideas either by using the current baseline models (and avoiding the integration burden) or by integrating their own systems into the platform.

Acknowledgments

This task is organised by the University of Sheffield (Loic Barrault), University of Le Mans (Fethi Bougares), Universitat Politècnica de Catalunya

(Marta R. Costa-jussà and Magdalena Biesialska) and LNE (Olivier Galibert) in the framework of the EU Chist-ERA funded ALLIES project. This work is is supported in part by the Spanish Ministerio de Ciencia e Innovación, through the postdoctoral senior grant Ramón y Cajal and by the Agencia Estatal de Investigación through the projects EUR2019-103819, PCIN-2017-079 and PID2019-107579RB-I00 / AEI / 10.13039/501100011033.

References

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A simple but tough-to-beat baseline for sentence embeddings. In *5th International Conference on Learning Representations, ICLR 2017*.

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. 2018. Unsupervised neural machine translation. In *6th International Conference on Learning Representations, ICLR 2018*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR*, arXiv:1409.0473.

Magdalena Biesialska, Katarzyna Biesialska, and Marta R. Costa-jussà. 2020. Continual lifelong learning in natural language processing: A survey. In *Proceedings of the 28th International Conference on Computational Linguistics*, Online. Association for Computational Linguistics.

Ozan Caglayan, Mercedes García-Martínez, Adrien Bardet, Walid Aransa, Fethi Bougares, and Loïc Barrault. 2017. Nmtpy: A flexible toolkit for advanced neural machine translation systems. *Prague Bull. Math. Linguistics*, 109:15–28.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

M. Amin Farajian, Marco Turchi, Matteo Negri, and Marcello Federico. 2017. Multi-domain neural machine translation through unsupervised adaptation. In *Proceedings of the Second Conference on Machine Translation, WMT 2017, Copenhagen, Denmark, September 7-8, 2017*, pages 127–137. Association for Computational Linguistics.

Marcello Federico. 2018. Challenges in adaptive neural machine translation. In *Proceedings of the AMTA 2018 Workshop on Translation Quality Estimation and Automatic Post-Editing*, pages 207–242, Boston, MA. Association for Machine Translation in the Americas.

Robert M. French. 1999. Catastrophic forgetting in connectionist networks. *Trends in Cognitive Sciences*, 3(4):128 – 135.

Fábio Kepler, Jonay Trénous, Marcos Treviso, Miguel Vera, and André F. T. Martins. 2019. OpenKiwi: An open source framework for quality estimation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics–System Demonstrations*, pages 117–122, Florence, Italy. Association for Computational Linguistics.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Philipp Koehn and Josh Schroeder. 2007. Experiments in domain adaptation for statistical machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, StatMT '07, page 224–227, USA. Association for Computational Linguistics.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Unsupervised machine translation using monolingual corpora only. In *International Conference on Learning Representations*.

Xiaoqing Li, Jiajun Zhang, and Chengqing Zong. 2018. One sentence one model for neural machine translation. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*, Miyazaki, Japan. European Languages Resources Association (ELRA).

Ming Liu, Wray Buntine, and Gholamreza Haffari. 2018. Learning to actively learn neural machine translation. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 334–344, Brussels, Belgium. Association for Computational Linguistics.

Minh-Thang Luong and Christopher D. Manning. 2015. Stanford neural machine translation systems for spoken language domain. In *International Workshop on Spoken Language Translation*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of*

the 40th Annual Meeting of the Association for Computational Linguistics, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

German I. Parisi, Ronald Kemker, Jose L. Part, Christopher Kanan, and Stefan Wermter. 2019. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54 – 71.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 1310–1318, Atlanta, Georgia, USA. PMLR.

Álvaro Peris, Miguel Domingo, and Francisco Casacuberta. 2016. Interactive neural machine translation. *Computer Speech & Language*, 45.

Ofir Press and Lior Wolf. 2017. Using the output embedding to improve language models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 157–163, Valencia, Spain. Association for Computational Linguistics.

Yevhenii Prokopalo, Sylvain Meignier, Olivier Galibert, Loïc Barrault, and Anthony Larcher. 2020. Evaluation of lifelong learning systems. In *Language Resources and Evaluation (LREC)*, Marseille, France.

Rico Sennrich, Orhan Firat, Kyunghyun Cho, Alexandra Birch, Barry Haddow, Julian Hitschler, Marcin Junczys-Dowmunt, Samuel Läubli, Antonio Valerio Miceli Barone, Jozef Mokry, and Maria Nădejde. 2017. Nematus: a toolkit for neural machine translation. In *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 65–68, Valencia, Spain. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Lucia Specia, Carolina Scarton, and Gustavo Henrique Paetzold. 2018. Quality estimation for machine translation. *Synthesis Lectures on Human Language Technologies*, 11(1):1–162.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958.

A LLMT system

This section describes the four different blocks that compose the LLMT system. The architecture of the system has been developed according to standard MT architectures. In order to facilitate the development of your system and to provide a baseline, a complete implementation of a LLMT system using nmtpytorch (Caglayan et al., 2017) is provided on the evaluation web page, see http://www.statmt.org/wmt20/lifelong-learning-task.html for more details.

General note: the prototypes of the **process** functions **must not** be changed!

A.1 Train and apply preprocessing

This block is responsible for preparing the training data. Preprocessing may include tokenization, learning subword decomposition model, etc. It is also responsible for creating the source and target vocabularies that will be used by the system. To do so, the entire training set is available at once (as in standard training protocol). The prepared training data is sent to the **train initial model** (sec. A.2) block while the subword model and vocabularies are sent to the **apply preprocessing** block (sec. A.3).

```
def process(self, data_loaders, outputs):
        # Get the training data
        data_loader = data_loaders[0]
        for i in range(data_loader.count()):
                (data, _, end_index) = data_loader[i]
                ... data["train_source_raw"].text
                ... data["train_target_raw"].text
                ... data["train_file_info"]
        #Note: setup_for_nmtpytorch(data_loaders) does that for you

        #HERE: DO AS MUCH DATA PREPARATION AS YOU WISH

        #Create vocabulary and BPE or SPM model
        data_dict_tok, src_vocab, trg_vocab, subword_model =
                preprocess(data_dict, self.source_language, self.target_language,
                        self.min_freq, self.short_list)

        data_dict_pickle = pickle.dumps(data_dict_tok).decode("latin1")

        #Write all the necessary outputs
        outputs['train_data_tokenized'].write({'text':data_dict_pickle}, end_index)
        outputs['source_vocabulary'].write({'text':src_vocab}, end_index)
        outputs['target_vocabulary'].write({'text':trg_vocab}, end_index)
        outputs['subword_model'].write({'text':subword_model}, end_index)

        # always return True, it signals BEAT to continue processing
        return True
```

A.2 Train initial model

The initial training of the system is implemented in the file *algorithms/loicbarrault/mt_train_model/1.py*. The process method is the main one. From this method, you can access all the training data from the **train_preprocessing** block. This block outputs a model.

```
# this will be called each time the sync'd input has more data available to be processed
def process(self, data_loaders, outputs):
        (data, _,end_data_index) = data_loaders[0][0]
        data_dict = pickle.loads(data["train_data"].text.encode("latin1"))

        #HERE: USE YOUR SOFTWARE FUNCTIONS TO TRAIN A MODEL

        # The model is Pickled with torch.save() and converted into a 1D-array of uint8
        # Pass the model to the next block
        outputs['model'].write({'value': model}, end_data_index)

        # always return True, it signals BEAT to continue processing
        return True
```

The data is available through the **data_loader**. In the provided baseline system, the processing consists of: tokenizing the data with Moses tokenizers (Koehn et al., 2007), training and applying a BPE model with subword_nmt (Sennrich et al., 2016b). As for the previous block, the **output** is written in the corresponding variable.

A.3 Apply preprocessing

The *apply preprocessing*'s algorithm is defined in the **process** function of the algorithm in *algorithms/loicbarrault/mt_apply_preprocessing/1.py* . The aim is to preprocess the lifelong data similarly to the training data using the vocabularies and subword models trained in the *train preprocessing* block.

The documents from the lifelong learning corpus are provided one after the other in the **input** parameter. Other information from previous blocks is available from the **data_loaders** as before.

```
# this will be called each time the sync'd input has more data available to be processed
def process(self, inputs, data_loaders, outputs):
  #Get the information from previous block,
  #NOTE: this should be done only once and stored in instance variable
  if self.src_bpe is None or self.trg_bpe is None \
        or self.src_vocab is None or self.trg_vocab is None:
    (data, _, end_data_index) = data_loaders[0][0]
    #Source and target vocabularies from the train_preprocessing block
    self.src_vocab = data["source_vocabulary"].text
    self.trg_vocab = data["target_vocabulary"].text
    #Source and target BPE objects to separate text into subwords units
    subword_model = io.StringIO(data["subword_model"].text)
    self.src_bpe = BPE(subword_model, vocab=self.src_vocab)
    self.trg_bpe = BPE(subword_model, vocab=self.trg_vocab)

  # Accessing lifelong data, one document at a time
  lifelong_source_raw = inputs['lifelong_source_raw'].data.text
  lifelong_target_raw = inputs['lifelong_target_raw'].data.text

  #HERE: APPLY THE PREPROCESSING TO THE DOCUMENT
  lifelong_source_tok = ...
  lifelong_target_tok = ...

  #Write all the necessary outputs
  outputs['lifelong_source_tokenized'].write({'text':lifelong_source_tok})
  outputs['lifelong_target_tokenized'].write({'text':lifelong_target_tok})
  if not inputs.hasMoreData():
    # DO SOMETHING WHEN ALL THE LIFELONG DATA HAS BEEN PROCESSED

  # always return True, it signals BEAT to continue processing
  return True
```

A.4 Lifelong learning loop

This block receives the initial model from the mt_train_initial_model block (sec. A.2) and process all files from the lifelong dataset provided by the *apply preprocessing* block, one at a time. This block has access to the whole training dataset and may store every processed document in memory in order to re-use it for further adaptation and/or any processing of your choice.

The output of this block is the translated document. This hypothesis might be obtained by simply translating the source document with the actual model (this is what the baseline model does). Eventually, you will plug your favorite unsupervised/semi-supervised or supervised adaptation scheme to create a better model before translating the document.

This module has also access to the user simulation (sec. A.5) from which the system can get reference translation for some segments in order to provide the best possible output.

A.5 User simulation

This module simulates the human in the loop. It receives requests from your system and provides answers to them. The requests and messages to the human are implemented in the *lifelong loop* block as dictionaries as follows:

```
request = {
  "request_type": "reference",
  "file_id": '{}'.format(file_id),
  "sentence_id": np.uint32(0)
}

message_to_user = {
  "file_id": file_id,  # ID of the file the question is related to
  "hypothesis": current_hypothesis[request['sentence_id']] ,
# The current hypothesis
  "system_request": request,  # the question for the human in the loop
}
```

As for now, only one type of request is available, namely 'reference'. This asks the user simulation to provide a correct translation for sentence number *sentence_id* from document *file_id*.

The answers are also a dict (see below) and can be obtained with the *validate* method as follows.

```
answer = {
  "answer": {"value": self.reference.text[sent_id]},
  "response_type": "reference",
  "file_id": self.file_info.file_id,
  "sentence_id": sent_id
 }
#Get the answer from the user simulation
human_assisted_learning, user_answer = loop_channel.validate(message_to_user)
```

Asking for human assistance is not free and will result in a penalisation of the system score, as described in sec. 3.2.

B How to setup a local platform for system development

B.1 Install

Installing the system requires to have a working conda[1] environment.

Then, the baseline system is available in the following repository: https://github.com/loicbarrault/allies_llmt_beat. Simply install using the *install.bash* script

B.2 Data

The data is available here: https://github.com/loicbarrault/allies_llmt_data. Simply follow the guidelines to recreate the data.

Update the *root_folder* at the bottom of the file *allies_llmt_beat/beat/databases/allies-mt-internal/1.json* with the path to the repository *allies_llmt_data/¡language-pair¿* directory (replace ¡language-pair¿ by the desired language pair, i.e. en-fr or en-de).

B.3 Run

Run the English→French system with the following command:

```
beat --prefix /path/to/git/allies_llmt_beat/beat exp run loicbarrault/loicbarrault/translation_ll_dev/1/translation_ll_dev
```

Run the English→German system with the following command:

```
beat --prefix /path/to/git/allies_llmt_beat/beat exp run loicbarrault/loicbarrault/translation_ll_dev/2/translation_ll_dev
```

[1] https://docs.conda.io/en/latest/

Findings of the WMT 2020 Shared Task on Chat Translation

M. Amin Farajian[1*] **António V. Lopes**[1*] **André F. T. Martins**[1,3]
Sameen Maruf[2] **Gholamreza Haffari**[2]
[1]Unbabel, Rua Castilho 52, 1250-069, Lisbon, Portugal
[2]Monash University, VIC, Australia
[3]Instituto de Telecomunicações, Instituto Superior Técnico, Lisbon, Portugal
{amin, antonio.lopes, andre.martins}@unbabel.com
{sameen.maruf, gholamreza.haffari}@monash.edu

Abstract

We report the results of the first edition of the WMT shared task on Chat Translation. The task consisted of translating bilingual conversational text, in particular customer support chats for the English-German language pair (English agent, German customer). This task varies from the other translation shared tasks, i.e. news and biomedical, mainly due to the fact that the conversations are bilingual, less planned, more informal, and often ungrammatical. Furthermore, such conversations are usually characterized by shorter and simpler sentences and contain more pronouns.

We received 14 submissions from 6 participating teams, all of them covering both directions, i.e. En→De for *agent* utterances and De→En for *customer* messages. We used automatic metrics (BLEU and TER) for evaluating the translations of both agent and customer messages and human document-level direct assessments to evaluate the agent translations.

1 Introduction

Despite the significant progress in Neural Machine Translation (NMT) in the last years (Vaswani et al., 2017; Hassan et al., 2018), most systems still operate at sentence-level, disregarding the context of previous sentences. It has been pointed out that ignoring the context may degrade the quality of translations, leading to incorrect choice of pronouns, lexical inconsistency, and incoherence (Läubli et al., 2018; Toral et al., 2018). This is particularly relevant in the context of bilingual chat translation, which normally consists of short messages, referencing each other, and where the correct lexical choice to translate a speaker might have been uttered in a previous turn by the other speaker.

Numerous systems have been proposed recently to address document-level translation (Tiedemann and Scherrer, 2017; Zhang et al., 2018; Maruf et al., 2019; Miculicich et al., 2018; Voita et al., 2019b; Tu et al., 2018; Maruf et al., 2018; Jean et al., 2017; Voita et al., 2018, 2019a; Junczys-Dowmunt, 2019; Lopes et al., 2020), focusing on extending both Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) and Transformer (Vaswani et al., 2017) with additional encoders or decoders to incorporate previous sentences context. However, often, the approaches are developed for single speaker and document-like tasks. By contrast, in this shared task, we focus on the online multispeaker and multi-lingual setting, where each participant in the conversation speaks in their native language. This task has been first considered by Maruf et al. (2018).

In the first round of the Chat Translation shared task, we propose translating dialogues with two speakers, where the first speaker is speaking in the German→English direction and the second is speaking in the English→German. Moreover, we tailor this task for a specific use case: translating conversational text of the customer support chats. In this setting the utterances of the German speaking customer are translated using a machine translation system into English. Then, the replies of the English speaking agent are translated into German and sent to the customer.

Translating conversational text, in particular customer support chats, is an important and challenging application task for machine translation technology. This type of content has so far not been extensively explored in prior MT research, largely due to the lack of publicly available data sets. Prior related work has mostly focused on movie subtitles and European Parliament speeches. To alleviate this problem, we created a corpus for this shared task, ***BConTrasT*** (§2), which is translated from English into German and is based on the monolingual Taskmaster-1 corpus (Byrne et al., 2019).

[*]These authors contributed equally.

Proceedings of the 5th Conference on Machine Translation (WMT), pages 65–75

The main motivation of this shared task is to analyze the challenges posed by conversational data as a content type, which has a broad application in industry-level services. In this content type, the text is usually not carefully well formatted, frequently contains typos, abbreviations, and inconsistent casing, usually with shorter sentences, often informal and ungrammatical. Since chat sessions are interactive, the task of translating conversations can be seen as a two-in-one task, modelling both dialogue and document-level translation at the same time.

In order to evaluate the translation quality of the participating systems we use both automatic metrics (BLEU (Papineni et al., 2002) and TER (Snover et al., 2006)), and human evaluation, consisting of Direct Assessment (DA). For DA, we define the evaluation process similarly to last year's WMT News Translation task (Barrault et al., 2019) with document-level context and following the set of recommendations of Läubli et al. (2020). However, differently than the News task, here we rely on professional translators instead of a crowd. This is mainly based on the observations of Läubli et al. (2020), which provides evidence of the professional translators having better judgment and ability to detect fine-grained phenomena.

Six teams participated in this first campaign of the Chat Translation shared task, with 14 runs in total. All teams submitted both English→German and German→English directions. In §4, we describe each system in more details.

2 Bilingual Conversational Data

One of the main challenges of bilingual conversation translation is the lack of publicly available data sets targeted for the task. The most commonly used datasets are movie subtitles (Lison and Tiedemann, 2016), European Parliament speeches (Koehn, 2005), and conversations extracted from the public forums such as Ubuntu Dialogue corpus (Lowe et al., 2015). These corpora, however, usually involve more than two speakers, contain a significant amount of noise (e.g. speakers information missing in the case of movie subtitles), and usually cover very broad domains.

For the Chat Translation task, we aim to develop a common ground for MT researchers to train and test their solutions by providing common training, validation, and test sets, as well as a common shared task definition. Unfortunately, due to the General Data Protection Regulation (GDPR), most commercial enterprises cannot distribute publicly their proprietary data. Therefore, we opted for using the Taskmaster-1 corpus (Byrne et al., 2019), which includes monolingual (English) task-based dialogues in six domains: (i) ordering pizza, (ii) creating auto repair appointments, (iii) setting up ride service, (iv) ordering movie tickets, (v) ordering coffee drinks, and (vi) making restaurant reservations. We used this corpus for creating the data of our shared task.

Since the main goal of this task is to enable multilingual speakers communicate with each other in their native language, we used the Unbabel translation service [1] to translate the utterances of both speakers into the target language (German). In this process, the conversations (originally in English) were first automatically translated into German and then manually post-edited by Unbabel editors, who are native German speakers. Having the conversations in both languages allows us to simulate bilingual conversations in which one speaker, the *customer*, speaks in German and the other speaker, the *agent*, answers in English. Table 1 shows the first few sentences of a bilingual conversation, along with their corresponding translations. In order to provide a realistic environment in which the amount of in-domain parallel data is scarce, we translated only a small set of the Taskmaster-1 corpus. Since pronouns are one of the main challenges in translating conversational data, we selected the conversations that contain at least one English anaphoric pronoun *it*. For this we used NEURALCOREF [2] and selected around 18k sentence pairs and then divided them into train, development, and test sets (see Table 2).

3 Task Description

A critical challenge faced by international companies today is delivering customer support in several different languages. One solution to this challenge is centralizing support with English speaking agents and having a translation layer in the middle to translate from the customer's language into the agent's (English) and vice versa. The ideal solution for this environment needs to consider the context of both sides which are in different languages, and also needs to be robust to the noisy input since the text here represents a higher degree of noise com-

[1] www.unbabel.com

[2] https://github.com/huggingface/neuralcoref

agent	src: Hi there! How can I help? tgt: Hallo! Wie kann ich helfen?
customer	src: Hey, ich muss mein Auto zum Mechaniker bringen und ich würde gerne Intelligent Auto Imports besuchen. tgt: Hey there, I need to take my car to mechanic and I would like to see Intelligent Auto imports.
agent	src: Sure! what type of car is it? tgt: Sicher! Was für ein Auto ist das?

Table 1: An example of a conversation between a customer and an agent.

	Customer		Agent	
	lines	words	lines	words
Training	6,216	41,492	7,629	70,193
Dev	862	5,805	1,040	9,569
Test	967	6,464	1,133	10,187

Table 2: Statistics of the English side of the training, dev, and test sets.

pared to the cases like news, biomedical, etc. In the first edition of this shared task we focused on this environment and asked the participants to translate the customer's utterances from German into English and the agent's from English into German.

Although participants were encouraged to submit both directions (i.e. modelling both speakers was desired), in this first round of the task, we emphasized on the agent side (English→German) and performed human evaluation in that direction exclusively. This decision is not entrenched and, thus, for future tasks we will aim at evaluating both translation directions. We decided to pursue this direction because the customer side (German→English) suffers from "translationese": English was the original source, and it was recently shown that translationese has a significant impact in evaluation both in automatic metrics (Freitag et al., 2020) and human evaluation (Läubli et al., 2020).

3.1 Data

The main data source for this shared task is ***BConTrasT***. As mentioned in §2, the translated conversations are sampled from the original Taskmaster-1 corpus, and in theory the other monolingual data could be leveraged by the participants either for back-translation or training in-domain language models. However, due to the high degree of sentence similarity within the Taskmaster-1 monolingual corpus, participants were not allowed to use this additional data to train their systems.

In addition to the provided in-domain training data, the participants were allowed to use all the training data provided by the News shared task organizers. Moreover, they were allowed to use existing pre-trained models, such as BERT (Devlin et al., 2018), Transformer-XL (Dai et al., 2019), Reformer (Kitaev et al., 2020), among others.

3.2 Baseline

To define our non-human baseline, we use Facebook's last year submissions to the document-level translation task for both directions (Ng et al., 2019) as the terms of comparison. Even though these models are not domain adapted for the Chat Translation task, we find them to have a reasonable quality for this domain. However, it is worth mentioning that we solely report the results of these models with the automatic metrics and we do not perform any type of direct assessment on these models.

4 Participants

Six participants submitted their systems to the Chat Translation shared task. Although the German→English direction (i.e. customer side) was optional, all participants submitted their systems for both directions. In total, 14 runs were submitted (although only primary submissions were considered for human evaluation). Table 3 summarizes the participants and their affiliations.

Team	Institution
NaverLabs	Naver Labs Europe
UEdinUppsala	Univ. of Edinburgh, Uppsala Univ.
IndTaoWang	Individual participant (Tao Wang)
Tencent	Tencent
UMaryland	University of Maryland
UJordan	Jordan U. of Science and Technology

Table 3: The participating teams and their affiliations.

4.1 Systems

Here we briefly detail each participant's systems as described by the authors and refer the reader to the participant's submission for further details.

4.1.1 Naver Labs

Naver Labs Europe (NLE) uses a document-level model trained on both the parallel and back-translated data. The authors developed a multi-domain system using the task-specific adapter layers and used it to participate in all the following tasks: chat translation, robustness, and biomedical. These systems are designed to translate both German and English text, or even mixed-language documents. Furthermore, in order to improve the robustness of these systems to noise, the authors applied the following pre-processing solutions: special handling of case with inline casing, a `copy` placeholder for rare characters, synthetic noise generation, and BPE dropout. Their primary submission is an ensemble of three instances of this model, which was used to decode the full bilingual dialogues at once using the entire dialogue's context. The first contrastive submission is a single model with these settings. The second submission is an ensemble of four sentence-level bidirectional models (one of them with masked language model pre-training). For more details see Bérard et al. (2020).

4.1.2 Universities of Edinburgh and Uppsala

The joint submissions of University of Edinburgh and Uppsala University are based on the transformer-big architecture (Vaswani et al., 2017) and rely on fine-tuning pre-existing systems from the WMT 2019 News Translation Task (experiment with both UEdin's submission based on Marian (Junczys-Dowmunt et al., 2018) and Facebook's submission based on Fairseq (Ott et al., 2019)). They are fine-tuned on pseudo-in-domain web crawled data and in-domain task data. The authors also experiment with (i) domain and speaker-level adaptation by automatically tagging the source and target sentences with domain and speaker tags respectively, and (ii) contextual NMT by exploiting the previous context, varying the type and number of previous utterances used. The final submission is an ensemble of four models trained with domain tags and using noisy-channel re-ranking. For more details see (Moghe et al., 2020).

4.1.3 Tao Wang (individual participant)

Individual participant Tao Wang uses a sentence-level system trained on all the WMT20 En-De parallel data. The author uses the Fairseq codebase to train a transformer-big model with the default settings of a base model. Then, the models are fine-tuned with the in-domain training set provided for the Chat Translation shared task.

4.1.4 Tencent

Tencent systems are based on self-attention networks including document-level multi-encoder and sentence-level Transformer. In order to get more in-domain data the authors use a multi-feature data selection method (e.g. FDA, n-gram LM, Transformer LM and BERT) to select data from news corpus. Furthermore, the systems have different fine-tuning strategies, ranging from sentence-level to document-level. Finally, these systems use large scale pre-trained language models including monolingual BERT (Devlin et al., 2018) and bilingual XLM (Lample and Conneau, 2019). For more details see (Wang et al., 2020).

4.1.5 University of Maryland

The University of Maryland systems are both sentence and document-level systems, with two distinct architectures for this task: (i) standard transformer pre-trained on WMT17 News and fine-tuned on the WMT20 Chat data, and (ii) modified transformer by including additional encoder to process one previous utterance in tandem with the current utterance, also pre-trained on WMT17 News and fine-tuned on a mix of WMT20 Chat data and a subset of WMT19 News data. The primary system is based on the first architecture while the second architecture is used for the two contrastive submissions. The contrastive submissions differ in the manner and timing in which training data was processed. For more details see (Bao et al., 2020).

4.1.6 Jordan University of Science and Technology

Mohammed et al. (2020) train separate models for the agent and customer sides after combining the training and development datasets for each side. They use bidirectional RNN (LSTM) with pre-trained BERT (Devlin et al., 2018) embeddings for each of the translation directions. In addition, the authors report using different parameters for training, resulting in different models which then are used for ensemble decoding. For more details see (Mohammed et al., 2020).

4.2 Submission Summary

The submissions for this year's shared task cover different approaches from simple sentence-level to more complex document-level models with extra encoders and decoders to summarize the context

(i.e. previous sentences), and from single direction to bi-directional translations (i.e. jointly modelling both En→De and De→En directions). Moreover, they report different approaches for training their systems ranging from fine-tuning the existing models and using embeddings of the large pre-trained models such as BERT (Devlin et al., 2018) to training the models from scratch.

Not only the submissions are different in their architectures, but they also differ in the data they use during the training. Some use all the available WMT parallel data in addition to the in-domain training data provided for the Chat task, and some apply data selection methods to get more in-domain data to leverage for training their systems.

5 Evaluation Procedures

For the first round of the Chat Translation shared task we follow the standard procedure of WMT shared tasks and evaluate both on automatic metrics and human evaluation with context. Even though automatic metrics provide a cheap mechanism to evaluate Machine Translation (MT) systems outputs, they do not tell the whole story for high-performing systems (Ma et al., 2019). For example, recent "sentence-level human parity" claims do not seem to hold when the context of the document is considered (Läubli et al., 2018), and metrics such as BLEU (Papineni et al., 2002) fail to correlate properly with human assessment (Callison-Burch et al., 2006). In this edition of the shared task, we aim for both automatic and manual evaluations.

5.1 Automatic Evaluation

For the automatic evaluation, we use both BLEU (Papineni et al., 2002) and TER (Snover et al., 2006) metrics. For the former, we use SacreBLEU[3] (Post, 2018), while for TER we use v0.7.25[4] and report case-sensitive scores. The automatic metrics are used to measure the quality of the translations of both sides, i.e. customer and agent.

5.2 Human Evaluation

For the human evaluation we follow a similar procedure to last year's WMT News shared task (Barrault et al., 2019) but take into account the set of recommendations defined by Läubli et al. (2020).

[3]BLEU+case.mixed+lang.en-de+numrefs.1+smooth.exp+tok.13a+version.1.4.13, BLEU+case.mixed+lang.de-en+numrefs.1+smooth.exp+tok.13a+version.1.4.13

[4]http://www.cs.umd.edu/~snover/tercom/

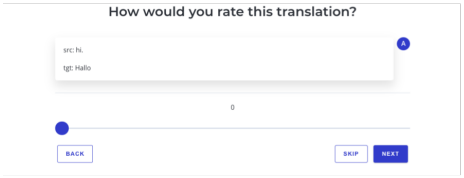
	Agent		Customer	
System	**BLEU↑**	**TER↓**	**BLEU↑**	**TER↓**
FAIR WMT'19	43.4	38.0	49.7	32.0
Primary				
NaverLabs	60.1	25.7	61.0	23.3
UEdinUppsala	60.2	25.4	**62.4**	**22.8**
IndTaoWang	59.7	26.0	61.3	23.5
Tencent	58.6	26.7	62.3	23.0
UniMaryland	56.7	28.2	49.4	32.0
UJordan	46.4	38.2	42.5	40.2
Contrastive				
NaverLabs-Sys1	58.8	26.8	59.4	24.6
NaverLabs-Sys2	**60.4**	**25.1**	61.6	23.1
UEdinUppsala-Sys1	60.2	25.3	61.8	22.8
UEdinUppsala-Sys2	59.8	25.4	61.5	23.8
Tencent-Sys1	53.6	30.6	54.0	28.8
Tencent-Sys2	58.6	26.6	61.9	23.2
UniMaryland-Sys1	55.6	28.3	49.4	32.0
UniMaryland-Sys2	56.4	28.1	49.4	32.0

Table 4: Automatic evaluation scores for the agent (En→De) and customer (De→En).

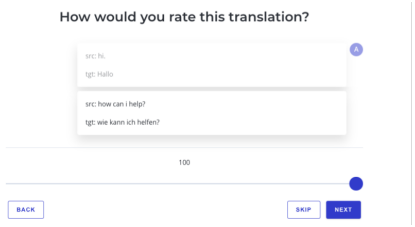
Specifically, we build *HITs* (following the Mechanical Turk's term *human intelligence task*) for the Segment Rating + Document Context (SR+DC) configuration with approximately 100 tasks similarly to WMT News, where both the source and target context is available to the evaluator when rating the actual source and target sentence for evaluation. We use an internal tool at Unbabel which provides the necessary visualization to evaluate a SR+DC configuration. Despite WMT News (Barrault et al., 2019) use Appraise (Federmann, 2012) for the human evaluation as it's tailored for document like text, the tool used for this task was built with chat evaluation in mind and outlines boundaries between each speaker. Figure 1 illustrates the tool used for evaluation.

Following Läubli et al. (2020) guidelines, we use trusted professional translators from the Unbabel community to evaluate the adequacy of the translation on a scale of 0 to 100. The guidelines to the translators were as simple as possible to avoid any type of bias, asking them to rate each sentence taking the context into account and penalizing when there is a context error, as they would for a non-contextual error.

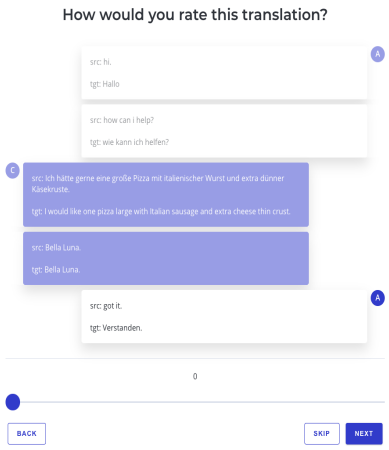
For the first edition of this shared task, we per-



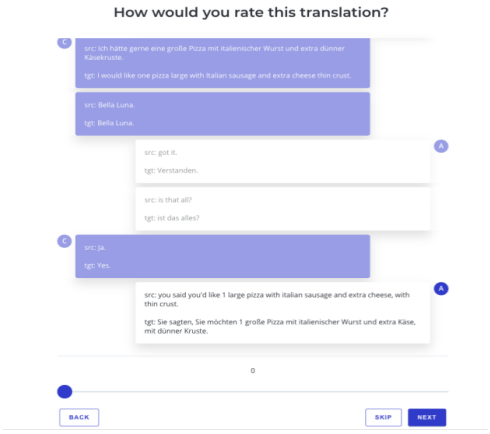
(a) First sentence of the conversation.



(b) Second sentence of the conversation.



(c) Third sentence of the agent in the conversation.



(d) Fifth sentence of the agent in the conversation.

Figure 1: Screenshots of a segment-rating with document-level context using the direct assessment tool. Multiple screenshots are presented to illustrate the iterative nature of the evaluation and how the agent and customer directions are presented as the conversation flows. Note that only the agent side is assessed and the scores are just illustrative.

formed human assessment on the agent side exclusively. Our decision is due to a limitation in the process of data creation, the *customer* direction is from professionally translated German (yet translated nonetheless) to the noisy original English (e.g. typos). Therefore, if we proceed with the evaluation as it stands we would induce two biases, 1) assessing a softer version of *translationese* as the source would be a translation, and 2) the noisy reference could bias the evaluators to rank the systems higher due to the noise and not quality. Both biases could be misleading and impacting their evaluations as professional translators are more sensitive to fine-grained phenomena (Läubli et al., 2020; Barrault et al., 2019). Moreover, in the proposed setting the impact of the noisy context for the agent is negligible for them to have a gist of the message; however there is an extra responsibility in translating the agent since the application of these systems in industry carries an additional factor: it has the company brand associated. Therefore, we preferred to focus more on evaluating the agent translations more rigorously than to spend resources in evaluating the customer.

5.2.1 Protocol for building HITs

We follow a hybrid between WMT News and Läubli et al. (2020) to build HITs. Specifically, as we resorted to professional translators there are fewer control tasks in every 100 HITs (i.e. 5% of the tasks being control tasks). To create a control task, we take inspiration from both the aforementioned resources and perform the following, assuming there is a vocabulary containing all the target words of the conversation: For the very short sentences containing one or two words we replace their words with some random words from the conversation's vocabulary. In the case of sentences with three words we replace the second and third words as before while keeping the first word. Finally, for longer sentences we preserve the first and last 10% of the words while randomly reordering the remaining 80% of the middle words. It is also worth mentioning that the corruption is only employed in the current sentence for evaluation and the context is preserved with no change.

When building the HIT bundle, among different options, we followed the same approach as WMT19 New's (Barrault et al., 2019) procedure

for SR+DC: in order to save time of our annotators, we built the HITs such that a sentence belonging to a given document is displayed and rated before the next sentence of the same document for the same participant MT system output. This is specially suited for our task as the conversations have larger contexts via numerous interactions. Similarly to WMT19 News (Barrault et al., 2019), we randomly picked documents from the pool of documents and for each participant retrieved their translations of that document. Next, we randomly picked documents from the pool until the sum of all their sentences was approximately 95 and added the remaining control tasks. For each document in the HIT, we sliced the translated conversation so that the order of the sentences was preserved when presented to the annotator for the SR+DC evaluation.

5.2.2 Evaluated Dialogs

Due to constraints with the annotators, we evaluated a subsample of the full test set. Therefore, we followed the procedure in § 5.2.1 with a budget constraint, where we specified the number of desired sentences and randomly sampled dialogues until the threshold is met (number of sentences). In the end, we evaluated 40% of the *agent* side, as noted in §5.2 we evaluated only this direction.

6 Discussion

The results of the automatic scores of both agent and customer side of all the submitted systems are reported in Table 4. Comparing these scores with our baselines (i.e. FAIR WMT'19 models) shows that in the agent side (En→De) there is a significant difference (i.e. between +3.0 to +17.0 BLEU scores) in the performance of the submitted systems and the baseline. However, comparing the differences between their TER scores reveals that there is a smaller gap between the systems, ranging from +0.2 to -12.9.

On the customer side we observe different behaviours and more diverse scores. In fact, the differences of the BLEU scores of the baseline and the submissions vary from -7.2 up to +12.7. This means that in a few cases our submitted systems fall behind the baseline by -7.2 BLEU scores. The TER scores show a similar behaviour and the differences of the scores of the submitted systems with the baseline varies from +8.2 (in the worst case) to -9.2 (in the case of best performing system). Given the fact that our references for this direction (i.e De→En) contain a higher degree of noise (eg. typos, wrong casings, etc) it is difficult to make a final and strong conclusion for this direction. We plan to investigate this aspect further.

	Agent	
System	**Avg.↑**	**Avg. z.↑**
Human	**91.4**	**0.319**
NaverLabs	88.2	0.165
UEdinUppsala	85.4	0.032
IndTaoWang	83.6	-0.049
UniMaryland	79.3	-0.235
Tencent	74.3	-0.474
UJordan	63.9	-0.966

Table 5: Human evaluation scores of the agent side.

Table 5 depicts the human evaluation scores (**Avg.**) and the normalized z-scores (**Avg. z**) of the agent side of the primary submissions. Human performance estimates are analogous to Barrault et al. (2019), evaluation of human-produced reference translations are denoted by "HUMAN" in all tables. There are three main clusters of scores, very high scores near human baseline levels (*NaverLabs*, *UEdinUppsala*, and *IndTaoWang*), significant scores (*UniMaryland* and *Tencent*), and lower scores (*UJordan*). Focusing on the high performing systems, we see that *NaverLabs* is the clear winner of the task, followed closely by *UEdinUppsala*, and *IndTaoWang*.

In addition to the overall DA scores of the submissions one might ask how they perform on the more detailed aspects such as sentences with different lengths or sentences containing pronouns. In order to address the first question, we analyzed the human scores for each system with respect to different intervals of lengths (i.e., different bins), namely 1-5 words, 6-10 words, 11-15 words, and, finally, 16+ words. To this end we can condition either (i) on the source sentence, or (ii) on the reference sentence, or (iii) on the generated translations of each system. Among these, we focused on (i) which provides more insights and is fairer comparison for all the systems.

Table 6 presents the human evaluation scores (**Avg.**) and the normalized z-scores (**Avg. z**) of the evaluated submissions in each length range. As we see, all the systems perform similarly in this range, all of them very close to the human reference. It is intersting to note that the submission of *UJordan* outperforms the human reference by +2.5

System	Source length range (words)							
	1-5		6-10		11-15		16+	
	Avg.	Avg. z.	Avg.	Avg. z.	Avg.	Avg. z.	Avg.	Avg. z.
Human	92.5	0.375	**92.5**	**0.367**	**90.0**	**0.254**	85.0	0.012
NaverLabs	92.5	0.375	86.9	0.103	88.3	0.170	**90.0**	**0.234**
UEdinUppsala	92.5	0.375	85.6	0.047	86.7	0.086	65.0	-0.936
IndTaoWang	92.5	0.360	83.1	-0.068	81.7	-0.146	75.0	-0.432
UniMaryland	90.0	0.249	79.4	-0.226	80.0	-0.210	55.0	-1.350
Tencent	85.0	0.042	71.9	-0.586	76.7	-0.378	65.0	-0.906
UJordan	**95.0**	**0.486**	71.3	-0.617	41.7	-2.012	10.0	-3.528

Table 6: Human evaluation scores of the agent side in each length range, based on the source sentences. The systems are ordered based on their general rankings.

	Agent	
System	**DA↑**	**z-score↑**
Human	**95.0**	**0.706**
NaverLabs	85.0	0.220
UEdinUppsala	85.0	0.220
Tencent	80.0	0.043
IndTaoWang	80.0	0.043
UniMaryland	80.0	0.043
UJordan	60.0	-0.861

Table 7: Human evaluation scores for the agent side when there is a pronoun *it* in the source sentence.

and +0.111 points on the average and normalized z-score, respectively. The differences increase by moving to the longer source sentences which is expected. The only unusual observation in these scores is the higher scores of the NaverLabs in the last range (i.e. sentences with 16+ words) in which it outperforms the human reference by +5.0 and +0.222 points on the average and normalized z-score, respectively. This can be due to the evaluators preferences, but still needs further analysis before making any final conclusion.

The English sentences containing pronouns is another aspect that we analyzed further and compared the performances of the submitted systems when there is a pronoun in the sentence. Specifically, we compute the scores for sentences which contain at least one instance of pronoun *it*. Table 7 shows the human scores and the normalized z-scores. As the results show, there is a big difference in the scores obtained by human translators and the submitted systems. In fact, it varies from -10.0 to -50.0 in the case of average score and from -0.486 to -1.567 for the normalized z-scores. Even though the number of tasks is not large, these preliminary results suggest current document-level systems still fall behind humans in challenging linguistic phenomena such as translating pronouns, and require further research for these phenomena.

Finally, we note that three of the submitted primary systems do not leverage the document-level context and use only the sentence-level information. Due to the data size and content proposed for the first edition of the Chat Translation shared task, this is to be expected as there is some level of repetition and similarity among different conversations. However, by looking at the results, we notice that approaches with document-level context seem to benefit from human evaluation when compared to the automatic metrics.

7 Conclusions

We presented the results of the first edition of the WMT20 Chat Translation shared task. For the purpose of this task, we created a bilingual English-German dialogue corpus, ***BConTrasT***, which is publicly available on the website of the task. It is based on the monolingual Taskmaster-1 corpus (Byrne et al., 2019) which was originally created in English. We translated around 18k of conversations of this corpus into German using the professional translators and used it as the in-domain corpus of the shared task.

This year we received 14 submissions from 6 different teams, all of them covering both directions (i.e. *customer* and *agent*). In addition to the automatic metrics (i.e. BLEU and TER) we performed an extensive Direct Assessment with document-level context using professional translators and used the results of these manual evalua-

tions to rank the participating systems. The previous sentences of each conversion provide the annotators with more context to have a more reliable assessment of the translations. Due to the constraints posed by our data, this year we were able to perform the manual evaluation only on the agent side (i.e. En→De). However, we aim at assessing both sides in the futures tasks.

Acknowledgments

We would also like to thank Mathieu Giquel and Ulisses Ferreira for all their help and support during the human evaluation phase, as well as Courtney Stankey for helping in coordinating with the evaluators. This work was supported by the P2020 programs MAIA (contract 045909) and Unbabel4EU (contract 042671), by the European Research Council (ERC StG DeepSPIN 758969), and by the Fundação para a Ciência e Tecnologia through contract UID/50008/2019.

References

Calvin Bao, Yow-Ting Shiue, Chujun Song, Jie S. Li, and Marine Carpuat. 2020. The university of maryland's submissions to the wmt20 chat translation task: Searching for more data to adapt discourse-aware neural machine translation. In *Proceedings of the Fifth Conference on Machine Translation*.

Loïc Barrault, Ondřej Bojar, Marta R Costa-Jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, et al. 2019. Findings of the 2019 conference on machine translation (wmt19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 1–61.

Alexandre Bérard, Vassilina Nikoulina, Ioan Calapodescu, and Jerin Philip. 2020. Naver labs europe's participation in the robustness, chat, and biomedical tasks at wmt 2020. In *Proceedings of the Fifth Conference on Machine Translation*.

Bill Byrne, Karthik Krishnamoorthi, Chinnadhurai Sankar, Arvind Neelakantan, Ben Goodrich, Daniel Duckworth, Semih Yavuz, Amit Dubey, Kyu-Young Kim, and Andy Cedilnik. 2019. Taskmaster-1: Toward a realistic and diverse dialog dataset. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4506–4517.

Chris Callison-Burch, Miles Osborne, and Philipp Koehn. 2006. Re-evaluation the role of bleu in machine translation research. In *11th Conference of the European Chapter of the Association for Computational Linguistics*.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988, Florence, Italy. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Christian Federmann. 2012. Appraise: An open-source toolkit for manual evaluation of machine translation output. *The Prague Bulletin of Mathematical Linguistics*, 98:25–35.

Markus Freitag, David Grangier, and Isaac Caswell. 2020. Bleu might be guilty but references are not innocent. *arXiv preprint arXiv:2004.06063*.

Hany Hassan, Anthony Aue, Chang Chen, Vishal Chowdhary, Jonathan Clark, Christian Federmann, Xuedong Huang, Marcin Junczys-Dowmunt, William Lewis, Mu Li, et al. 2018. Achieving human parity on automatic chinese to english news translation. *arXiv preprint arXiv:1803.05567*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Sebastien Jean, Stanislas Lauly, Orhan Firat, and Kyunghyun Cho. 2017. Does neural machine translation benefit from larger context? *arXiv preprint arXiv:1704.05135*.

Marcin Junczys-Dowmunt. 2019. Microsoft translator at wmt 2019: Towards large-scale document-level neural machine translation. In *Proceedings of the Fourth Conference on Machine Translation*.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia. Association for Computational Linguistics.

Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. 2020. Reformer: The efficient transformer. In *International Conference on Learning Representations*.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT Summit*. Citeseer.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *Advances in Neural Information Processing Systems (NeurIPS)*.

Samuel Läubli, Sheila Castilho, Graham Neubig, Rico Sennrich, Qinlan Shen, and Antonio Toral. 2020. A set of recommendations for assessing human–machine parity in language translation. *Journal of Artificial Intelligence Research*, 67:653–672.

Samuel Läubli, Rico Sennrich, and Martin Volk. 2018. Has machine translation achieved human parity? a case for document-level evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4791–4796, Brussels, Belgium. Association for Computational Linguistics.

Pierre Lison and Jörg Tiedemann. 2016. Opensubtitles2016: Extracting large parallel corpora from movie and tv subtitles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 923–929.

António V Lopes, M Amin Farajian, Rachel Bawden, Michael Zhang, and André F T Martins. 2020. Document-level Neural MT: A Systematic Comparison. In *22nd Annual Conference of the European Association for Machine Translation*, pages 225–234, Lisboa, Portugal.

Ryan Lowe, Nissan Pow, Iulian Vlad Serban, and Joelle Pineau. 2015. The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 285–294.

Qingsong Ma, Johnny Wei, Ondřej Bojar, and Yvette Graham. 2019. Results of the WMT19 metrics shared task: Segment-level and strong MT systems pose big challenges. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 62–90, Florence, Italy. Association for Computational Linguistics.

Sameen Maruf, André F. T. Martins, and Gholamreza Haffari. 2019. Selective attention for context-aware neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3092–3102, Minneapolis, Minnesota. Association for Computational Linguistics.

Sameen Maruf, André FT Martins, and Gholamreza Haffari. 2018. Contextual neural model for translating bilingual multi-speaker conversations. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 101–112.

Lesly Miculicich, Dhananjay Ram, Nikolaos Pappas, and James Henderson. 2018. Document-level neural machine translation with hierarchical attention networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2947–2954, Brussels, Belgium. Association for Computational Linguistics.

Nikita Moghe, Christian Hardmeier, and Rachel Bawden. 2020. The university of edinburgh-uppsala university's submission to the wmt 2020 chat translation task. In *Proceedings of the Fifth Conference on Machine Translation*.

Roweida Mohammed, Mahmoud Al-Ayyoub, and Malak Abdullah. 2020. Just system for wmt20 chat translation task. In *Proceedings of the Fifth Conference on Machine Translation*.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook fair's wmt19 news translation task submission. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and Ralph Weischedel. 2006. A study of translation error rate with targeted human annotation. In *In Proceedings of the Association for Machine Transaltion in the Americas (AMTA 2006*.

Jörg Tiedemann and Yves Scherrer. 2017. Neural machine translation with extended context. In *Proceedings of the Third Workshop on Discourse in Machine Translation*, pages 82–92, Copenhagen, Denmark. Association for Computational Linguistics.

Antonio Toral, Sheila Castilho, Ke Hu, and Andy Way. 2018. Attaining the unattainable? reassessing claims of human parity in neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers, WMT 2018, Belgium, Brussels, October 31 - November 1, 2018*, pages 113–123. Association for Computational Linguistics.

Zhaopeng Tu, Yang Liu, Shuming Shi, and Tong Zhang. 2018. Learning to remember translation history with a continuous cache. *Transactions of the Association for Computational Linguistics*, 6:407–420.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Elena Voita, Rico Sennrich, and Ivan Titov. 2019a. Context-aware monolingual repair for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 877–886, Hong Kong, China. Association for Computational Linguistics.

Elena Voita, Rico Sennrich, and Ivan Titov. 2019b. When a good translation is wrong in context: Context-aware machine translation improves on deixis, ellipsis, and lexical cohesion. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1198–1212, Florence, Italy. Association for Computational Linguistics.

Elena Voita, Pavel Serdyukov, Rico Sennrich, and Ivan Titov. 2018. Context-aware neural machine translation learns anaphora resolution. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*.

Longyue Wang, Zhaopeng Tu, Xing Wang, Li Ding, Liang Ding, and Shuming Shi. 2020. Tencent AI Lab machine translation systems for the WMT20 chat translation task. In *Proceedings of the Fifth Conference on Machine Translation*.

Jiacheng Zhang, Huanbo Luan, Maosong Sun, Feifei Zhai, Jingfang Xu, Min Zhang, and Yang Liu. 2018. Improving the transformer translation model with document-level context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 533–542, Brussels, Belgium. Association for Computational Linguistics.

Findings of the WMT 2020 Shared Task on Machine Translation Robustness

Lucia Specia[1], **Zhenhao Li**[1], **Juan Pino**[2], **Vishrav Chaudhary**[2], **Francisco Guzmán**[2]
Paul Michel[3], **Graham Neubig**[3], **Hassan Sajjad**[4], **Nadir Durrani**[4], **Yonatan Belinkov**[5],
Philipp Koehn [2,6], **Xian Li**[2]
[1]Imperial College London, [2]Facebook AI, [3]Carnegie Mellon University,
[4]Qatar Computing Research Institute, [5]Harvard University and MIT, [6]Johns Hopkins University

Abstract

We report the findings of the second edition of the shared task on improving robustness in Machine Translation (MT). The task aims to test current machine translation systems in their ability to handle challenges facing MT models to be deployed in the real world, including domain diversity and non-standard texts common in user generated content, especially in social media. We cover two language pairs – English-German and English-Japanese and provide test sets in zero-shot and few-shot variants. Participating systems are evaluated both automatically and manually, with an additional human evaluation for "catastrophic errors". We received 59 submissions by 11 participating teams from a variety of types of institutions.

1 Introduction

In recent years, Machine Translation (MT) systems have seen great progress, with neural models becoming the *de-facto* methods and even approaching human quality in news domain (Hassan et al., 2018). However, like other deep learning models, neural machine translation (NMT) models are found to be sensitive to synthetic and natural noise in input, distributional shift, and adversarial examples (Koehn and Knowles, 2017; Belinkov and Bisk, 2018; Durrani et al., 2019; Anastasopoulos et al., 2019; Michel et al., 2019). From an application perspective, MT systems need to deal with non-standard, noisy text of the kind which is ubiquitous on social media and the internet, yet has different distributional signatures from corpora in common benchmark datasets.

Following the first shared task on Machine Translation (MT) Robustness, we now propose a second edition, which aims at testing MT systems' robustness towards domain diversity. Specifically, this year's task aims to evaluate a general MT system's performance in the following two scenarios:

- Zero-shot: the goal is to evaluate a general MT system's performance in unseen domains at test time, which are likely to be different from a training domain (e.g. News, Wikipedia). For that, no domain-specific data or information on the test sets is given to participants.

- Few-shot: the goal is to test an MT system's performance if a few in-domain training examples are provided for the target domain. The question we ask is: can the general MT system leverage those training examples to improve performance on this domain while not dropping its performance on other domains?

We describe the dataset and the task setup in Section 3. The shared-task attracted a total of 23 submissions from 11 teams. The teams employed a variety of methods to improve robustness. A specific challenge was the small size of the in-domain noisy parallel dataset. We summarize the participating systems in Section 4 and some trends on approaches used by various systems in Section 4.1. The contributions were evaluated both automatically and via a human evaluation and the results discussed in Section 5.

We hope that this task leads to more efforts from the community in building robust MT models.

2 Related Work

Domain mismatch is a key challenge in machine translation (Koehn and Knowles, 2017). Most approaches for improving robustness of MT systems to domain shift assume the existence of significant amounts of parallel data in both the source and target domain. In this scenario, a common approach is to first train an MT system on a (generic) source domain and then to fine-tune it on a (specific) target domain (Luong and Manning, 2015; Freitag and Al-Onaizan, 2016; Servan et al., 2016; Chu

Proceedings of the 5th Conference on Machine Translation (WMT), pages 76–91

et al., 2017), to continuously fine-tune on datasets increasingly similar to the target domain (Sajjad et al., 2017), or to dynamically change the balance of data towards the target domain (van der Wees et al., 2017). Another approach trains a system on multiple domains at the same time, while adding domain-specific tags to the input examples (Kobus et al., 2016). Both these approaches were employed by participants of the first shared task on MT robustness (Li et al., 2019).

Other methods for domain adaptation of MT systems include instance weighting (Wang et al., 2017b), incorporating a domain classifier (Chen et al., 2017; Britz et al., 2017), and data selection (Wang et al., 2017a). Some make use of monolingual data available either in the target domain—for example by training the decoder on such data (Domhan and Hieber, 2017) or by back-translating it (Sennrich et al., 2016)—or in the source domain, via similar techniques (Zhang and Zong, 2016).

Chu and Wang (2018) provide a broad survey of domain adaptation for neural MT, which demonstrates that the predominant setup assumes knowledge of the target domain and availability of target domain data at training time. In light of this prior work, the shared task proposed a relatively under-explored scenario, where examples in the target domain are either unavailable or relatively few.

Other aspects of robustness are robustness to adversarial examples or noisy inputs. The fragility of neural MT models has been previously demonstrated in various settings (Belinkov and Bisk, 2018; Heigold et al., 2017; Anastasopoulos et al., 2019; Lee et al., 2018). Michel and Neubig (2018) proposed a new dataset (MTNT) to test MT models for robustness to the types of noise encountered in the Internet. The previous iteration of the shared task focused on robustness of MT systems to such noise (Li et al., 2019). We refer to that report for more details.

3 Task

To facilitate comparability with the News translation task and also to reduce the participation cost, we suggest the same training data as the WMT20 News task.[1] The focus of the Robustness Task is to both evaluate models built on this type of data on more challenging test sets, as well as to encourage participants to explore novel training and modeling approaches so that models have more robust performance at test time on multiple domains, including unseen and diversified domains. We offer two language pairs: English-German (En→De) and English-Japanese (En→Ja), with different test sets focusing on one or both these language pairs, or one particular language direction.

3.1 Phases

The test cycle is divided into two phases. In the first phase – **zero-shot phase**, we release blind test sets from a mixture of domain(s), and participants submit their system's output without any information on these blind domains or training/development data for them. In the second phase – **few shot phase**, we release a small amount of training data (10K sentence pairs) from one of the test domains and participants submit their system's output after utilizing these training examples.

3.2 Training Data

The task includes two tracks, *constrained* and *unconstrained* depending on whether the system is trained on a predefined training datasets or not. The two tracks are evaluated by the same automatic and human evaluation protocol, however, they are compared separately.

- **Constrained:** Participants can only use the training data made available for this year's News translation task for training. They can use both the parallel data and monolingual data provided in this year's task. Multilingual systems trained with data provided by WMT20 News task are also allowed (and participants should indicate whether this is the case).

- **Unconstrained:** Participants can develop novel solutions to learn from unlabelled data, especially additional monolingual data from domains such as biomedical and/or Reddit. The online systems that we evaluated also fall in this category.

- **Few-shot:** Participants are provided a few in-domain training examples. The data provided consist of the German-English train and valid portions of the CoVoST dataset (deduplicated by source German sentences) and the Japanese-English and English-Japanese

[1] http://www.statmt.org/wmt20/translation-task.html

train and valid portions of the MTNT dataset (Michel and Neubig, 2018).

3.3 Development Data

The task specified the following data to help participants evaluate their system's performance on unseen and multiple domains.

- English-German: participants can use the development data from the News translation task, development data from QED (Abdelali et al., 2014) corpus, development data from WMT19 Medical translation task, and development data from the WMT16 IT translation task.

- English-Japanese: participants can use the development data from the News translation task, and development data from the MTNT dataset, which contains noisy social media texts and their clean translations.

3.4 Test Data

We have three test sets which were created using different sources and approaches. The general statistics are reported in Table 1.

Wikipedia Comments Test Set (set1): This data was collected by Imperial College London and Facebook. We created this to be a particularly challenging test set where the source segments contain various types of linguistic constructs that could lead to what we call *catastrophic errors* in the MT output. For that, we chose user-generated content, namely comments on Wikipedia edits by Wikipedia editors. More specifically, we took English Wikipedia comments from an existing dataset from the Toxic Comment Classification Challenge.[2] The Challenge made available 160,000 comments on Wikipedia edits tagged with multi-grade toxicity labels (toxic, severe toxic, obscene, threat, insult, or identity hate). We believe that the presence of toxic content can be very challenging for MT systems.

After filtering out non-English segments and segments that were too long (>50 words or >1000 characters) or too short (<5 words), we kept all the remaining comments with any toxic label (approx. 7K) and randomly selected 10K non-toxic samples.

Based on this initial selection of 17K English comments, we defined heuristics to further sample from the selection and diversify the potential sources of catastrophic errors. To that end, we first machine translated all comments using an in-house transformer-based model into Japanese and German. The goal of that was to be able to examine potential differences in source and (one example of) translation segments.[3] We then pre-processed and automatically annotated all 17K segments with the following soft labels for catastrophic errors:

1. Introduction of toxicity: we checked both source and machine translation for toxic words (using in-house lists) and labelled as positive (i.e. potentially containing errors) cases where the source does not contain such words, but the translation does (at least one).

2. Mistranslation of named entities: we annotated person, organisation and location named entities in the source and translation (using an in-house named entity recognition model) and labelled as positive cases where (a) the translation has fewer named entities than the source and the translation has at least one toxic word, (b) the translation has at least 2 fewer named entities than the source, and (c) the list of named entity types (e.g. person vs location) in source and translation differ and translation has at least one toxic word.

3. Inversion of sentiment: we applied the Google Cloud Sentiment Analysis tool[4] to annotate each source and machine translation and labelled as positive cases with very different sentiments, i.e. the source is very positive (>0.5) and the translation is very negative (<-0.5) or vice-versa (scores range from -1 (negative) to 1 (positive).

4. Difference in emojis: we detected emojis in the source and machine translation[5] and labelled as positive cases where source and translation have a different number of emojis.

[2] www.kaggle.com/c/jigsaw-toxic-comment-classification-\challenge

[3] We are aware that using one particular translation model can bias the selection to cases that are challenging for this particular model. In future work following this methodology, we recommend that multiple MT models be used.

[4] https://cloud.google.com/natural-language/docs/analyzing-sentiment

[5] https://github.com/carpedm20/emoji/)

5. Presence of idioms: we checked if the source contains idiomatic expressions, using an in-house list of idioms built from various sources, and labelled those cases as positive.

We note that the automatic labelling using our various pre-processing techniques may have introduced errors, but we believe that basing the selection on such heuristics will still lead to higher chances of selecting very challenging source segments than arbitrarily sampling the data.

We divided the original data (toxic and non-toxic 17K) into 5 sets, one for each of these soft labels (allowing for duplicates samples across sets). Finally, we uniformly selected a test set per language pair, containing 1,098 unique segments for English→German and 1,100 unique segments for English→Japanese. We provided the test sets for experiments in both directions, but we will only report results on the original source→target direction. For each of these test sets, we discarded the machine translation and collected reference translations from scratch using professional translators.

Reddit Test Set (set2): This data was collected by Carnegie Mellon University following the same procedure as last year's test set (described in Michel and Neubig (2018)): comments from the social media website `reddit.com` were scraped, filtered for noisy comments and translated by professional translators. This year, data was collected for two translation directions: English→Japanese and Japanese→English. For English, comments were collected from the `/r/all` feed, which encompasses all communities, and filtered for English. Since Japanese is a minority language on Reddit, comments were scraped from a selection of japanese-speaking communities, detailed in Michel and Neubig (2018).

Common Voices Test Set (set3): This data was obtained from from the CoVoST corpus (Wang et al., 2020). CoVoST is derived from Common Voice (Ardila et al., 2020), a crowdsourced speech recognition corpus with an open CC0 license. Transcripts were sent to professional translators and the quality of translations was controlled by automatic and manual checks (Guzmán et al., 2019). For this task, we used the German→English test set with source German sentences deduplicated.

3.5 Evaluation protocol

Automatic evaluation: We first computed BLEU (Papineni et al., 2002) for each system using SacreBLEU (Post, 2018). For all language pairs except En→Ja, we used the original reference and SacreBLEU with the default options. In the case of En→Ja, we used the reference tokenized with KyTea and the option `--tokenize none`.

Human evaluation: The system outputs were evaluated by professional translators. The translators were presented the original source sentence, the reference and the system output side by side. The order between the reference and the system output, as well as the different MT systems, was randomized and not disclosed to the translator. The translators rated both the reference and the translation. We believe that the reference translation in this evaluation setup to serves the purpose of calibration by offering the human annotators one (hopefully) good example of translation. We also report metrics for these reference translations as an upperbound for the data.

We sampled 400 translations from each MT system in each of the test sets and language pairs (28 combinations), resulting in 11,200 segments and their references to be annotated (22,400 segments in total). Each translation/reference segment was annotated by three raters. Quality control was manged by the company providing the ratings, where the main check was that the three ratings could not disagree by more than one category (in which case additional raters are enlisted until agreement is reached).

The rating of translations was done using a different metric from last year's task. Instead of direct assessment (DA), we chose a discrete *likert* rating ranging from 1 to 5, which we found to lead to higher agreement between raters in other annotation projects (Diab et al., 2020). A summary of the guidelines provided for this *likert* rating is as follows:

1 Bad: translation errors are so severe that they cause the target text to be incomprehensible. This may be mainly due to major grammatical, typographical, or lexical errors, or omission of critical or important salient information.

2 Poor: the target text contains translation errors, but these errors do not hinder overall comprehension and do not mistranslate overall intent.

	En→De	De→En	En→Ja	Ja→En
Wikipedia Comments (set1)	1,098 / 26,549	-	1,100 / 29,419	-
Reddit (set2)	-	-	1,376 / 20,011	997 / 20,842
Common Voice (set3)	-	5,609 / 43,119	-	-

Table 1: Number of sentences/words per test set (Japanese words are counted after tokenization with KyTea).

The errors may be mainly due to partial differences in intent, grammatical or typographical errors, or omission of important salient information.

3 Acceptable: the target text is fully comprehensible and fully translated (i.e. no information is omitted), even if it contains minor errors. These errors may be mainly due to partial lack of fluency, or a few grammatical or typographical errors.

4 Very Good: the target text is fully comprehensible, fully correct, and does not miss any information. Style matters may not be transferred faithfully, such as level of formality, or the translation of idioms does not need to be perfect but their meaning needs to be correctly conveyed.

5 Excellent: the target text is fully comprehensible, fully correct, and does not miss any information. Additionally, source style is reflected in the translation and if present, idioms are perfectly handled.

Catastrophic error annotation: As an additional form of human annotation, alongside the *likert* ratings described above, we instructed the annotators to indicate, for translations rated below **3 - poor or bad**, whether they contained any catastrophic errors, and to categorise the type of error. This is a new type of evaluation and we provided detailed guidelines, which we summarise below.

Annotators were asked to provide a **YES/NO flag** to indicate whether the translation contains any error (one or more words) that changes the meaning of the source segment in a critical way. Critical errors are those that lead to misleading translations which may carry religious, health, safety, legal or financial implications, or introduce toxicity. The set of critical errors used for the guidelines (which also included examples of these errors) includes – but is not limited to – the cases below:

- Introduction of toxicity (profanity, violence, hate or abuse) (TOX).
- Introduction of health/safety risks (SAF).
- Mistranslation of named entities (NAM).
- Reverse negation (NEG).
- Reverse of sentiment/polarity (SEN).
- Change in units/time/date/numbers (NUM).
- Other (OTH).

If the answer is YES, annotators were asked select one of the categories indicating the type of critical error. They were asked to choose the category that compromises the meaning of the sentence the most if more than one error was found in the same segment. Three raters flagged and categorized errors.

4 Participants and System Descriptions

We received submissions from 8 teams participating across different tasks, test sets and languages we provided this year. Below we briefly present the systems we were able to get a system description paper for:

Naver Labs (NLE): They participated in Chat and Biomedical tasks along with the Robustness task. They trained a general big-transformer model using *FairSeq* toolkit (Ott et al., 2019) and adapted it towards different tasks using lightweight adapter layers for each task (Bapna and Firat, 2019). They compared results against the more traditional fine-tuning method (Luong and Manning, 2015) to show that the former provides a viable alternative, while significantly reducing the amount of parameters per task. They also explored using embedding from pre-trained language models in their NMT system of which they tried two MLM variants: i) using NMT encoder's setting, using Roberta (Liu et al., 2019). The latter was found more robust to novel domains and noise. The authors found that initializing with first 8 layers instead of the entire model to

be optimal. Another notable finding included the use of single bidirectional model instead of monodirectional models to give similar performance. For the robustness task specifically they added source side synthetic noise and used BPE drop-out. While this was found to be useful in handling noisy data, no gains towards domain robustness were observed.

LIMSI: LIMSI participated in Biomedical and Robustness tasks. For the robustness challenge their main exploration was using adapter layers (Bapna and Firat, 2019) applied on 8 domains (parallel data released in the News task). The architecture adds an additional, domain-specific layer on top of every layer of the encoder and the decoder. This allows the test sets from known domains to use adapter layers and for novel domains to use the generic system. They created a noisy domain by adding synthetic noise to source data. The idea is that residual adapter layer learned from such data learns how to deal with noisy domain and is also able to preserve performance on the cleaner domains. However this did not work as well. The residual adapter fine-tuned using the ParaCrawl corpus gave better performance.

e-Translation: Their effort was mainly directed towards the News translation task, however they submitted two systems to the Robustness task. Their general systems were built using big-transformer configuration trained using Marian (Luong and Manning, 2015) after up-sampling original training data. The system was then fine-tuned for another round with an LM scored subset of original data. Finally ensembling four checkpoints produced their final systems. The authors reported an interesting finding that their models performed better on the noisy test sets released for this task than on the standard news test set, suggesting that systems trained on the diverse domains were already robust enough.

UEDIN: Team UEDIN also mainly trained their system towards News translation task, but added Gumbel noise to the output layer of the systems submitted to the Robustness task. They followed standard NMT training pipeline and boasted their systems with additional data filtered from the paracrawl corpus. The data was carefully selected using dual cross-entropy (Junczys-Dowmunt, 2018) and length-normalized cross-entropy.

OPPO: Team OPPO also trained their systems for the language pairs released for the News translation task and did not carry any specific exploration towards the task of Robustness. Their systems followed standard training regime of training transformer models with Marian toolkit, with back-translation to generate synthetic data and ensembles of models. As additional module, they added to their system a reranker trained on six forward and backward models, the scores of which are used as features in training the reranker.

PROMPT: Team PROMPT also participated mainly in the News translation task. Their systems were trained using OpenNMT (Klein et al., 2017) toolkit. They applied several stages of data preprocessing including length-based filtering, removing duplications, and using in-house classifier based on `Hunalign` aligner to identify and discard non-parallel sentences. They used two types of synthetic data to improve their systems: i) randomly selecting subset of Wikipedia equal to the size of news data and generating parallel corpus through back-translation, ii) creating synthetic data with unknown words using the procedure described in (Pinnis et al., 2017). Systems were trained with tags to differentiate between original data and synthetic data from each other. Named entities were handled through a post-processing module with re-decoding whenever a named entity was not translated or translated incorrectly.

Online systems: We also evaluated three top performing online MT systems, which are also commonly used in the WMT News translation task: online-A, online-B, and online-G. While we do not have access to details of the architectures of these models, to the best of our knowledge they are are all neural MT models with one case including a selection between translations from statistical and neural models.

4.1 Common Trends

Participating systems were trained following a standard recipe, i) using big-transformer models, ii) boasting performance with tagged back-translation, iii) continued training with filtered data and in-domain data (where available), iv) ensembling different models to obtain further improvements. Only two teams, namely Naver Labs and LIMSI made specific efforts towards the task of Robustness. Both of them used lightweight domain adaptors proposed by Bapna and Firat (2019). Both teams

also explored making the systems robust by adding noisy synthetic data. While they found using adaptor layers instead of fine-tuning the entire model to be a viable alternative, no success was observed adding noise to the training process.

5 Results

In this section we describe the results of both automatic and manual evaluation of general translation quality (Section 5.1), as well as an analysis of catastrophic errors (Section 5.2).

5.1 General Quality

Overall, the correlation between human judgments and BLEU is not strong. For En→De (set1), the Pearson's correlation coefficient is 0.97, while for the other four tasks the coefficients are lower, with 0.78, 0.65, 0.52, 0.79 for En→De (set1), Ja→En (set2), En→Ja (set2), and De→En (set3) respectively.

Automatic Evaluation The automatic evaluation (BLEU) results of the Shared Task are summarized in Table 2, where we also include the three online translation systems. We performed significance test using `compare-mt` (Neubig et al., 2019) where systems are considered as significantly different at $p < 0.05$. The result of significance test is used for the automatic evaluation ranking.

Overall, the **unconstrained** online-B system provides strong results and outperforms most systems in the five language pairs, except the De→En (set3) and En→Ja (set1).

Among the participating teams, the best **zero-shot** systems were OPPO, which outperforms other zero-shot systems in En→De (set1), Ja→En (set2), and En→Ja (set2) tasks, and NLE, which outperforms other systems in the other two tasks.

Only Naver Labs participated in the **few-shot** stage (NLE-few) and submitted their systems in four language directions except the En→De (set1) subtask. Their few-shot systems ranked the first in all the four directions they participated, tying online-B system in three language pairs.

Human Evaluation The results of human evaluation following the evaluation protocol described in Section 3.5 are outlined in Table 3. The *likert* score is calculated by averaging ratings from the three human annotators over the 400 sampled translations for each MT system, and we performed significance test using the `testSignificance.py` script[6] (Dror et al., 2018) with $p < 0.05$. The result of significance test in *likert* score is used for the human judgement ranking. Interestingly, the correlation in the system rankings between human judgments and BLEU is not strong. In other words, the best performing systems in BLEU do not rank high according human judgement, sometimes even rank the lowest. For example, in Ja→En (set2), the online-B system ranks first in BLEU but last in *likert* score. OPPO outperforms all systems in both directions on set2, and is overall the best system among the **constrained, zero-shot** submissions.

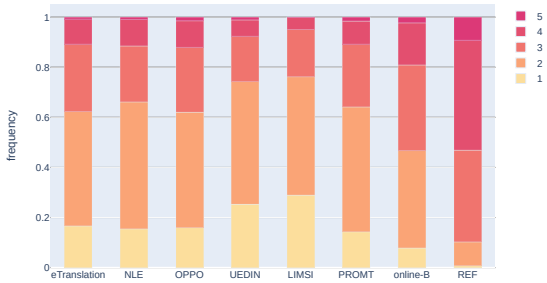
To get insight on the proportion of sentences with each of the categories of human score, Figure 1 displays the distribution of *likert* ratings for all systems. The most frequent ratings for the participating systems are 2 and 3 while for the human-translated references it is 4. Comparing the few-shot and zero-shot systems, the NLE-few outperforms most systems because the frequency of lower ratings (1 or 2) is lower, but the frequency of high ratings (5) is similar to the zero-shot systems.

5.2 Evaluation on Catastrophic Failures

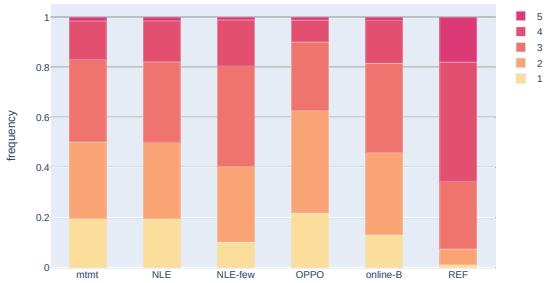
Here we turn our attention to the extra level of annotation where human raters flag and categorise catastrophic errors in sentences. We note that we had three raters for each translation, and that in some cases different categories of errors were flagged. This naturally happened since the raters were asked to choose the category with the biggest negative impact, which is a subjective decision. For example, in En→De (set1), each system has 28 sentences in average flagged with multiple errors. We report this average multi-error counts in Figure 3. In addition, we note that there may also be cases of disagreement, where only a subset of raters flag errors (we will perform agreement analysis later).

Error rate of systems Table 3 shows the proportion of sentences containing as least one error in (which we will refer as "error rate"). The error rates vary among different test sets. Regarding set1, which is sourced from Wikipedia comments, oversampling for more challenging content, the error rate for different systems is high, ranging from 51% to 76%. It is interesting that annotators indicate that the human-translated references contain catastrophic errors as well, with an error rate of 23% for both language pairs in set1. The error rate in

[6] https://github.com/rtmdrr/testSignificanceNLP

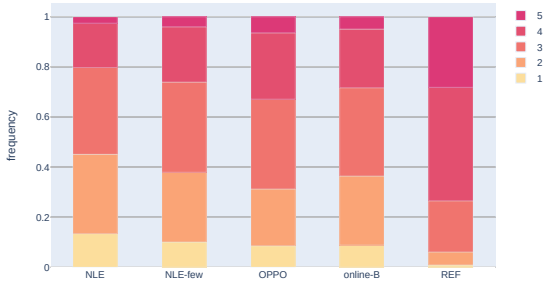


(a) set1: En→De

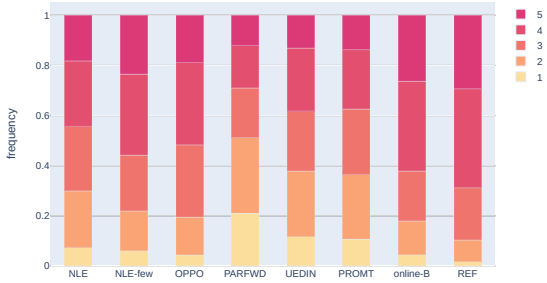


(b) set1: En→Ja

(c) set2: Ja→En



(d) set2: En→Ja



(e) set3: De→En

Figure 1: Distribution of *likert* ratings for all submitted systems (the darker the color, the higher the ratings - higher quality).

System	BLEU (RANK)				
	set1			set2	set3
	En→De	En→Ja	Ja→En	En→Ja	De→En
Constrained					
eTranslation	41.9 (3)	–	13.9 (2)	–	–
mtmt	–	18.2 (5)	–	–	–
NLE	42.2 (4)	22.5 (3)	13.3 (2)	16.2 (3)	44.7 (2)
NLE(FEW)	–	**25.4** (1)	**15.3** (1)	18.4 (1)	**45.4** (1)
OPPO	42.9 (2)	19.1 (5)	15.2 (1)	17.3 (2)	43.3 (3)
PARFWD	–	–	–	–	30.8 (5)
UEDIN	35.1 (7)	–	–	–	43.8 (3)
LIMSI	30.2 (8)	–	–	–	–
Unconstrained					
PROMT	41.4 (5)	–	–	–	41.4 (4)
online-A	38.6 (6)	23.1 (2)	13.6 (2)	17.8 (2)	43.2 (3)
online-B	**48.0** (1)	**25.4** (1)	14.3 (1)	**18.8** (1)	44.3 (2)
online-G	37.9 (7)	20.4 (4)	9.4 (3)	14.8 (3)	43.4 (3)

Table 2: Automatic evaluation (corpus-level BLEU, cased) over all submitted systems, with the system's rank in parentheses ($p < 0.05$). Bold highlights the system with highest BLEU score.

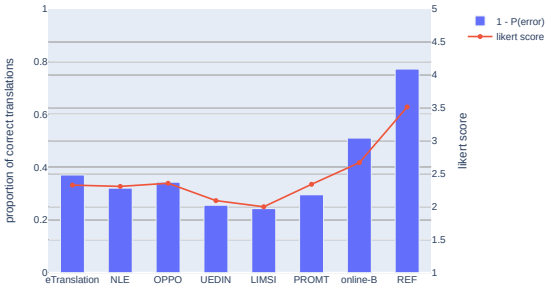
set2, sourced from Reddit, is lower, which is within 36%-51% for participating systems and 16%-18% for the references. In set3, which is sourced from Common Voice data, the error rate is the lowest. All systems except one achieve less than 10% error rate. The issue of catastrophic errors in the reference translations needs further investigation. We speculate that this could be due to misinterpretation of the guidelines, as we discuss below.

The error rate is highly correlated with the *likert* score reported in Section 5.1. We show in Figure 2 the relation of the proportion of translations without catastrophic errors (blue bars) and the *likert* scores (red lines). As expected, systems with more translations without errors get higher *likert* scores. The Pearson's correlation coefficient for De→En (set3) is 92%, while for the other four language pairs, the coefficients are over 96%.

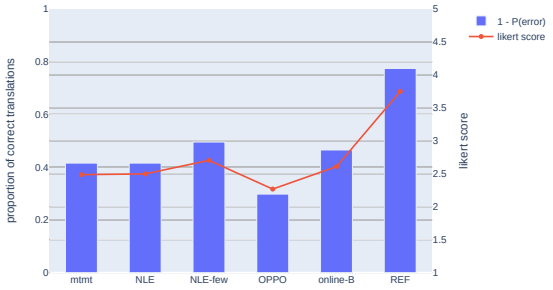
Distribution of error types In Figure 3 we show the absolute counts and proportion of different types of catastrophic errors per system. We note that some sentences may have been annotated with more than one error type (by different human annotators), and therefore the counts may seem inflated. To provide a better idea of the distribution of errors, for each system the error proportion is calculated as the number of translations with certain error divided by the number of sampled translations, i.e. 400. In all five language pairs, the OTH error is the main source of catastrophic errors, however, this OTH error is not clearly defined and might indicate different translation errors, e.g. some translations simply copy the source sentence and are therefore labelled as OTH error. This requires further analysis.

Excluding the OTH error (Figure 4), the catastrophic error distribution varies in different subtasks. Named entities (NAM) account for a large proportion of errors in all subtasks except En→De (set3). In En→De (set1), Ja→En (set2), and De→En (set3) subtasks, sentiment (SEN) errors are very frequent, similar to NAM errors. The TOX error is predominant only in En→Ja subtask. Other types of catastrophic errors occupy much smaller proportion.

This figure also highlights the different catastrophic error types flagged for reference translations. While this needs further inspection and investigation, we suspect that annotators might have misinterpreted the guidelines. For example, in the Wikipedia comments En→Ja, there is a large proportion of sentences with catastrophic errors of the type "toxic" (TOX): almost 10% of the reference translations contain such error type. Translations (human or machine) containing toxic content might have been tagged as containing errors, even though the source segments also contained such toxic content and the translation is simply transferring it.



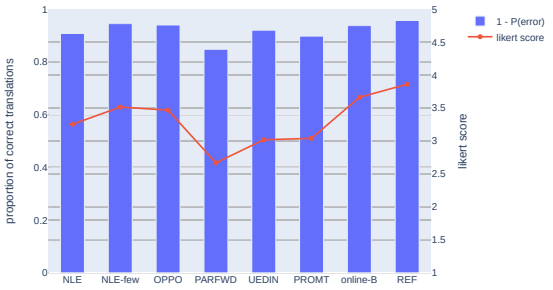
(a) set1: En→De



(b) set1: En→Ja

(c) set2: Ja→En

(d) set2: En→Ja



(e) set3: De→En

Figure 2: Proportion of translations without any error (bars) and *likert* over all submitted systems (red points/line).

System	*likert* score / error rate (RANK)				
	set1		set2		set3
	En→De	En→Ja	Ja→En	En→Ja	De→En
Constrained					
eTranslation	2.33 / 63% (2)	–	2.84 / 51% (1)	–	–
mtmt	–	2.49 / 59% (3)	–	–	–
NLE	2.31 / 69% (2)	2.50 / 59% (3)	2.74 / 49% (2)	2.64 / 49% (3)	3.25 / 9% (3)
NLE(FEW)	–	**2.70** / 51% (1)	2.87 / 46% (1)	2.82 / 36% (2)	3.51 / 6% (2)
OPPO	2.36 / 66% (2)	2.27 / 70% (4)	**2.93** / 45% (1)	**3.00** / 37% (1)	3.47 / 6% (2)
PARFWD	–	–	–	–	2.67 / 15% (5)
UEDIN	2.09 / 75% (3)	–	–	–	3.02 / 8% (4)
LIMSI	2.00 / 76% (4)	–	–	–	–
Unconstrained					
PROMT	2.34 / 71% (2)	–	–	–	3.04 / 10% (4)
online-B	**2.67** / 49% (1)	2.61 / 54% (2)	2.69 / 50% (2)	2.88 / 42% (2)	**3.66** / 6% (1)
Reference	3.51 / 23 %	3.75 / 23%	3.76 / 18%	3.95 / 16%	3.86 / 4%

Table 3: Average human judgments and catastrophic error translation rates over all submitted systems and the reference translations (p <0.05). The systems' rank for each translation direction is shown in parentheses. The best system is **highlighted**.

However, this would not explain other error types, which are defined in terms of mistranslation or mismatches between source and target content, such as incorrect named entity translation (NAM). We will analyse the data for that, as well as make it available.

6 Conclusions

The second edition of this WMT shared task focused on testing MT systems in more challenging conditions than last year, in two ways: (i) by making this in a zero-shot setting, where no training set and no in-domain development set were provided, (ii) by biasing the selection of the test sets to make them even harder to translate, for example, by oversampling segments with toxic content. We hoped to encourage participants in the other WMT translation tasks to submit to this task.

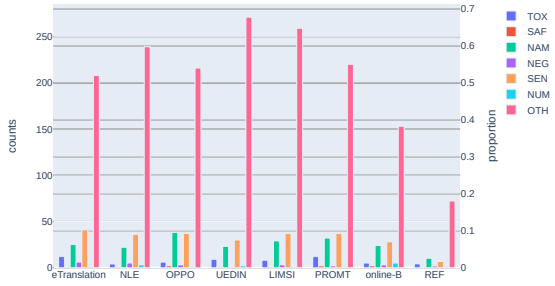
Indeed, most participating teams submitted standard NMT models trained on other types of data and other WMT tasks. Very few teams introduced specific techniques for robustness, such as augmenting training data with synthetic noise. Perhaps not entirely surprisingly, strong online systems, which are trained on a large variety of text types and domains, performed well according to both automatic and human evaluation. The only few-shot submission, however, managed to outperform online systems in most test sets, even in those from a different domain from the small training set provided. This is an interesting outcome and shows that few-shot settings are promising.

A new protocol was used for human evaluation: for general quality, direct assessment was replaced by *likert* scores with more detailed guidelines. The ranking of systems according to this human evaluation does not always agree with that given by BLEU, which is not surprising. According to human evaluation, systems were ranked together more often.

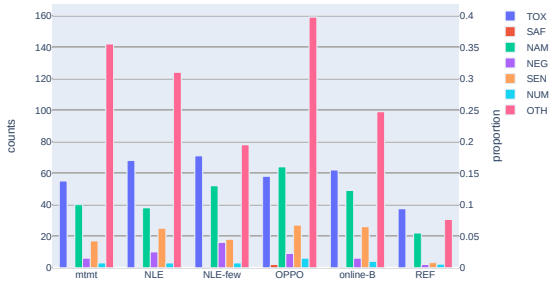
In addition to general quality, we also introduced a flag for catastrophic errors, which is a novel way to evaluate translations. The proportion of sentences containing such errors seems a lot higher than expected. This could be an artefact of the perception of human annotators on what constitutes a catastrophic error. This would explain why even the reference translations are found to contain such errors, albeit on a much smaller scale. In future work we will carry out in depth analysis on the annotation to investigate this high number of catastrophic errors in human and machine translations.

Acknowledgements

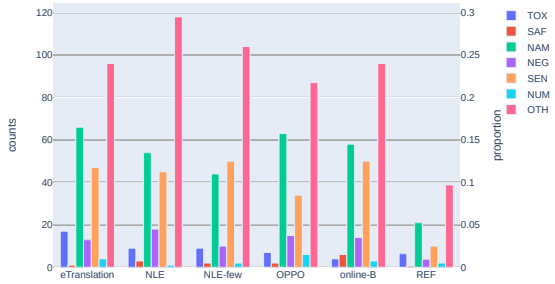
Lucia Specia was supported by funding from the Bergamot project (EU H2020 Grant No. 825303). We thank Facebook for funding the human evaluation. We thank Khetam Al Sharou for her help with



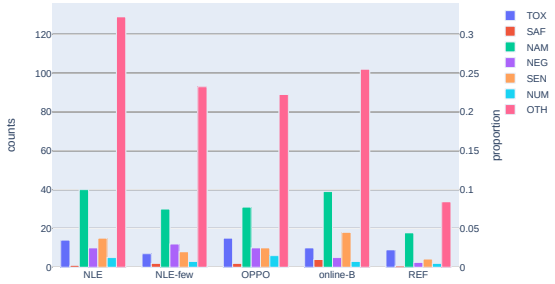
(a) set1: En→De (avg. multi-error sentences: 28)



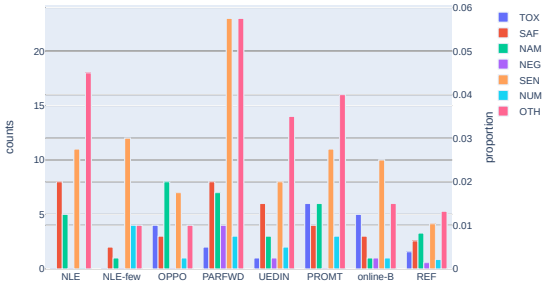
(b) set1: En→Ja (avg. multi-error sentences: 28)



(c) set2: Ja→En (avg. multi-error sentences: 24)

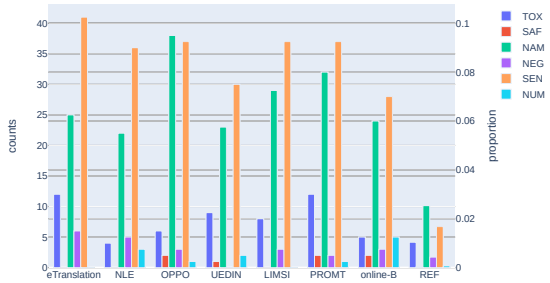


(d) set2: En→Ja (avg. multi-error sentences: 10)

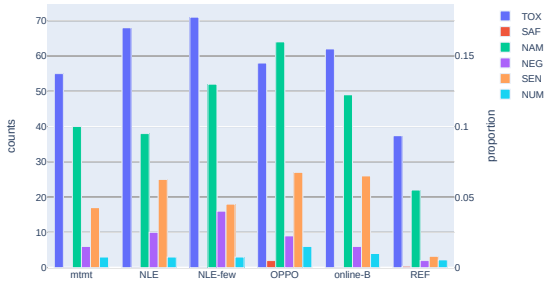


(e) set3: De→En (avg. multi-error sentences: 4)

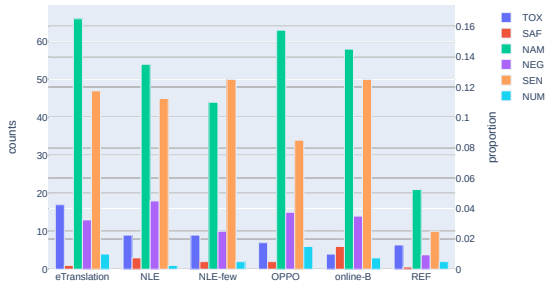
Figure 3: Distribution of different types of catastrophic errors for all systems: Absolute count or each error type per system, as well as proportion of sentences in each system that contain that error. The average number of sentences labelled with multiple errors per system is reported in parentheses.



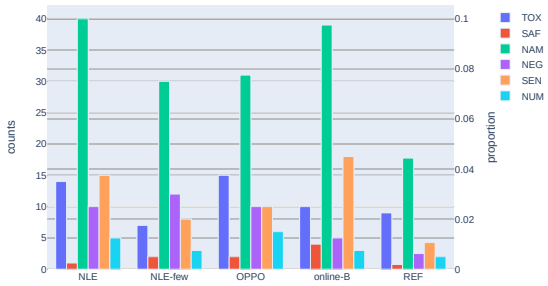
(a) set1: En→De



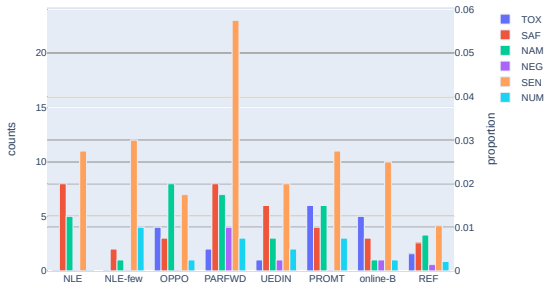
(b) set1: En→Ja



(c) set2: Ja→En



(d) set2: En→Ja



(e) set3: De→En

Figure 4: Distribution of different types of catastrophic errors for all systems **excluding OTH**: Absolute count or each error type per system, as well as proportion of sentences in each system that contain that error.

guidelines for catastrophic errors.

References

Ahmed Abdelali, Francisco Guzman, Hassan Sajjad, and Stephan Vogel. 2014. The AMARA corpus: Building parallel language resources for the educational domain. In *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC)*, Reykjavik, Iceland.

Antonios Anastasopoulos, Alison Lui, Toan Q. Nguyen, and David Chiang. 2019. Neural machine translation of text from non-native speakers. In *Proc. NAACL HLT*.

Rosana Ardila, Megan Branson, Kelly Davis, Michael Kohler, Josh Meyer, Michael Henretty, Reuben Morais, Lindsay Saunders, Francis Tyers, and Gregor Weber. 2020. Common voice: A massively-Âmultilingual speech corpus. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 4218–4222, Marseille, France. European Language Resources Association.

Ankur Bapna and Orhan Firat. 2019. Simple, scalable adaptation for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1538–1548, Hong Kong, China. Association for Computational Linguistics.

Yonatan Belinkov and Yonatan Bisk. 2018. Synthetic and natural noise both break neural machine translation. In *International Conference on Learning Representations (ICLR)*.

Denny Britz, Quoc Le, and Reid Pryzant. 2017. Effective domain mixing for neural machine translation. In *Proceedings of the Second Conference on Machine Translation*, pages 118–126, Copenhagen, Denmark. Association for Computational Linguistics.

Boxing Chen, Colin Cherry, George Foster, and Samuel Larkin. 2017. Cost weighting for neural machine translation domain adaptation. In *Proceedings of the First Workshop on Neural Machine Translation*.

Chenhui Chu, Raj Dabre, and Sadao Kurohashi. 2017. An empirical comparison of simple domain adaptation methods for neural machine translation. *CoRR*, abs/1701.03214.

Chenhui Chu and Rui Wang. 2018. A survey of domain adaptation for neural machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1304–1319, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Mona Diab, Denise Diaz, Ahmed Kishky, Anh Ngo, Ashley Chen, Paco Guzman, and Cynthia Gao. 2020. Rethinking direct assessment machine translation evaluation protocols for user-generated data: A comparative study. In *preparation*.

Tobias Domhan and Felix Hieber. 2017. Using target-side monolingual data for neural machine translation through multi-task learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1500–1505, Copenhagen, Denmark. Association for Computational Linguistics.

Rotem Dror, Gili Baumer, Segev Shlomov, and Roi Reichart. 2018. The hitchhiker's guide to testing statistical significance in natural language processing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1383–1392, Melbourne, Australia. Association for Computational Linguistics.

Nadir Durrani, Fahim Dalvi, Hassan Sajjad, Yonatan Belinkov, and Preslav Nakov. 2019. One size does not fit all: Comparing NMT representations of different granularities. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1504–1516, Minneapolis, Minnesota. Association for Computational Linguistics.

Markus Freitag and Yaser Al-Onaizan. 2016. Fast domain adaptation for neural machine translation. *CoRR*, abs/1612.06897.

Francisco Guzmán, Peng-Jen Chen, Myle Ott, Juan Pino, Guillaume Lample, Philipp Koehn, Vishrav Chaudhary, and Marc'Aurelio Ranzato. 2019. The FLORES evaluation datasets for low-resource machine translation: Nepali–English and Sinhala–English. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6098–6111, Hong Kong, China. Association for Computational Linguistics.

Hany Hassan, Anthony Aue, Chang Chen, Vishal Chowdhary, Jonathan Clark, Christian Federmann, Xuedong Huang, Marcin Junczys-Dowmunt, William Lewis, Mu Li, et al. 2018. Achieving human parity on automatic chinese to english news translation. *arXiv preprint arXiv:1803.05567*.

Georg Heigold, Günter Neumann, and Josef van Genabith. 2017. How robust are character-based word embeddings in tagging and mt against wrod scramlbing or randdm nouse? *arXiv preprint arXiv:1704.04441*.

Marcin Junczys-Dowmunt. 2018. Dual conditional cross-entropy filtering of noisy parallel corpora. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 888–895,

Belgium, Brussels. Association for Computational Linguistics.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Vancouver, Canada. Association for Computational Linguistics.

Catherine Kobus, Josep Maria Crego, and Jean Senellart. 2016. Domain control for neural machine translation. *CoRR*, abs/1612.06140.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. *arXiv preprint arXiv:1706.03872*.

Katherine Lee, Orhan Firat, Ashish Agarwal, Clara Fannjiang, and David Sussillo. 2018. Hallucinations in neural machine translation. In *Interpretability and Robustness in Audio, Speech, and Language Workshop Conference on Neural Information Processing Systems*.

Xian Li, Paul Michel, Antonios Anastasopoulos, Yonatan Belinkov, Nadir Durrani, Orhan Firat, Philipp Koehn, Graham Neubig, Juan Pino, and Hassan Sajjad. 2019. Findings of the first shared task on machine translation robustness. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 91–102, Florence, Italy. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Minh-Thang Luong and Christopher D. Manning. 2015. Stanford Neural Machine Translation Systems for Spoken Language Domains. In *Proceedings of the International Workshop on Spoken Language Translation*, Da Nang, Vietnam.

Paul Michel, Xian Li, Graham Neubig, and Juan Miguel Pino. 2019. On evaluation of adversarial perturbations for sequence-to-sequence models. In *Proc. NAACL HLT*.

Paul Michel and Graham Neubig. 2018. MTNT: A testbed for Machine Translation of Noisy Text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Graham Neubig, Zi-Yi Dou, Junjie Hu, Paul Michel, Danish Pruthi, and Xinyi Wang. 2019. compare-mt: A tool for holistic comparison of language generation systems. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 35–41, Minneapolis, Minnesota. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Marcis Pinnis, Rihards Krislauks, Daiga Deksne, and Toms Miks. 2017. Neural machine translation for morphologically rich languages with improved subword units and synthetic data. In *Text, Speech, and Dialogue - 20th International Conference, TSD 2017, Prague, Czech Republic, August 27-31, 2017, Proceedings*, volume 10415 of *Lecture Notes in Computer Science*, pages 237–245. Springer.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

Hassan Sajjad, Nadir Durrani, Fahim Dalvi, Yonatan Belinkov, and Stephan Vogel. 2017. Neural machine translation training in a multi-domain scenario. In *Proceedings of the International Workshop on Spoken Language Translation (IWSLT)*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Christophe Servan, Josep Maria Crego, and Jean Senellart. 2016. Domain specialization: a post-training domain adaptation for neural machine translation. *CoRR*, abs/1612.06141.

Changhan Wang, Juan Pino, Anne Wu, and Jiatao Gu. 2020. Covost: A diverse multilingual speech-to-text translation corpus. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 4197–4203, Marseille, France. European Language Resources Association.

Rui Wang, Andrew Finch, Masao Utiyama, and Eiichiro Sumita. 2017a. Sentence embedding for neural machine translation domain adaptation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Short Papers)*.

Rui Wang, Masao Utiyama, Lemao Liu, Kehai Chen, and Eiichiro Sumita. 2017b. Instance weighting for neural machine translation domain adaptation. In *Proceedings of the the Conference on Empirical Methods in Natural Language Processing,*.

Marlies van der Wees, Arianna Bisazza, and Christof Monz. 2017. Dynamic data selection for neural machine translation. In *Proceedings of the the Conference on Empirical Methods in Natural Language Processing*.

Jiajun Zhang and Chengqing Zong. 2016. Exploiting source-side monolingual data in neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545, Austin, Texas. Association for Computational Linguistics.

The University of Edinburgh's English-Tamil and English-Inuktitut Submissions to the WMT20 News Translation Task

Rachel Bawden Alexandra Birch Radina Dobreva
Arturo Oncevay Antonio Valerio Miceli Barone Philip Williams

School of Informatics, University of Edinburgh, Scotland
{rbawden,abirch,rdobreva,a.oncevay,amiceli,pwillia4}@ed.ac.uk

Abstract

We describe the University of Edinburgh's submissions to the WMT20 news translation shared task for the low resource language pair English-Tamil and the mid-resource language pair English-Inuktitut. We use the neural machine translation transformer architecture for all submissions and explore a variety of techniques to improve translation quality to compensate for the lack of parallel training data. For the very low-resource English-Tamil, this involves exploring pretraining, using both language model objectives and translation using an unrelated high-resource language pair (German-English), and iterative backtranslation. For English-Inuktitut, we explore the use of multilingual systems, which, despite not being part of the primary submission, would have achieved the best results on the test set.

1 Introduction

The University of Edinburgh participated in the WMT20 news translation shared task for English-Tamil and English-Inuktitut in both translation directions.[1,2] Neither language pair benefits from large quantities of parallel data, so we approach training using different techniques to compensate for this lack of data: pretraining and iterative backtranslation for English-Tamil and multilingual systems for English-Inuktitut. We use neural machine translation (MT) and specifically the transformer architecture (Vaswani et al., 2017): the base variant for the lower-resourced English-Tamil and the big variant for the mid-resource English-Inuktit. In both cases, significant improvements are seen when compared to the in-house baselines tested, particularly notable for pretraining for English-Tamil.

Awaiting the results of the official human evaluation, we report the automatic evaluation scores using BLEU (Papineni et al., 2002) as implemented in sacreBLEU (Post, 2018). A summary of these results on the dev and test sets can be found in Table 1 for all UEDIN submissions. The details of our submissions can be found in Section 2 for English-Tamil and in Section 3 for English-Inuktitut.

Language direction	Dev	Test
EN→TA	12.30	8.40
TA→EN	21.00	16.60
EN→IN	27.0	8.2
IN→EN	48.8	23.0

Table 1: Summary of results for all UEDIN submissions according to the automatic evaluation (BLEU).

2 English↔Tamil

As for our English↔Gujarati systems last year at WMT19 (Bawden et al., 2019), we use pretraining and data augmentation to tackle the low-resource language pair English–Tamil. Our experiments show that pre-training, training on backtranslated data and then fine-tuning is useful in both directions, although we introduce slight variations in the training and fine-tuning approaches used for each language direction.

2.1 Data and pre-processing

Our models are trained in the constrained scenario, using publicly available WMT20 data. We choose to exclude the terminology-like Wikititles as well as WikiMatrix[3] from our training data, using only

[1]The UEDIN participation for English-German is in a separate submission.

[2]Code and models can be found at http://data.statmt.org/wmt20_systems/.

[3]While term lists contain useful vocabulary, they can inundate the training data due to their large size. This can cause translation problems due to the different nature of the text, notably in terms of sentence length. The EN-TA portion of WikiMatrix corpus is very noisy adn so this is excluded too.

Proceedings of the 5th Conference on Machine Translation (WMT), pages 92–99
Online, November 19–20, 2020.

Data type	#sentences	Corpora
Parallel en–ta	340,995	PMindia, Tanzil, NLCP, PIB, MKB, EnTam
Monolingual en (in-domain)	653,606,835	News (crawl, discussions, commentary)
Monolingual en (out-of-domain)	101,692,093	Europarl, Wiki dumps
Monolingual ta (in-domain)	668,008	News crawl
Monolingual ta (out-of-domain)	1,553,160	Wiki dumps
Parallel de–en	43,675,462	Europarl, News commentary, Paracrawl, WikiMatrix, Tilde Rapid

Table 2: Data used for the Tamil-English models. Note that we also use German-English data for some of our experiments as a form of pretraining.

the corpora shown in Table 2. We use both parallel data and monolingual data for English-Tamil and also exploit parallel data available for English-German as a form of pre-training.

All data was first cleaned, keeping sentences of 3–100 (untokenised) units, for which the length ratio between the parallel sentences is maximum 2.2, and do not contain more than 50% non-alphabetic characters or more than 50% of words without an alphabetic character.[4] We deduplicate the data and normalise punctuation using Moses (Koehn et al., 2007). We then apply subword segmentation using SentencePiece (Kudo and Richardson, 2018) and the BPE strategy (Sennrich et al., 2016).[5]

2.2 Approach used

We adopt a three-step approach to training our models, consisting of: (i) *pre-training* model parameters using either an mBART language model or a translation model for the highly resourced De-En language pair, (ii) *iterative backtranslation* to produce synthetic parallel data of increasing quality, and (iii) *final model creation* consisting of fine-tuning pretrained models using both genuine parallel and backtranslated data. We provide the details of these three steps below.

Pre-training We experimented with several pre-training objectives: language modelling using XLM (Lample and Conneau, 2019a) or mBART (Liu et al., 2020), and MT pre-training using a higher-resourced language pair (namely English-German). Using a higher-resourced language pair for pretraining, even if this pair is unrelated to the language pair on which the model is fine-tuned, has shown to be an effective and simple way of boosting performance (Kocmi and Bojar, 2018; Aji et al., 2020). For the De-En models, we had to choose between (i) initialising only model parameters and (ii) preserving all model and training parameters from the parent model (similar to Grundkiewicz et al. (2019)). We chose the first option as it produced better results in our experiments.

For mBART pretraining, we use all Tamil and English monolingual data without shuffling or deduplication. We tag the input segments with a language tag and a domain tag: either in-domain (news) or out-of-domain as in (Caswell et al., 2019). For XLM pretraining we use the deduplicated and shuffled corpus (since cross-sentence context is not needed) and we subsample the English because of computing cost. We also use domain tags, with language information provided in the form of language embeddings as per the standard implementation. For De-En pre-training, we use all De-En parallel data described in Table 2, with a joint English-Tamil-German vocabulary. We experiment with pretraining models in the two directions (De→En and En→De) and find that the De→En model produces better results when fine-tuned on TA-EN data.

System	EN→TA		TA→EN	
	dev	test	dev	test
Parallel-only baseline	5.10	3.10	10.10	10.60
XLM	7.44	5.00	13.44	10.90
mBART	7.40	4.65	14.00	13.40
De-En	7.30	5.00	13.60	14.20

Table 3: Comparison of pre-training methods for EN↔TA (BLEU) after fine-tuning on parallel data.

Table 3 shows the results of each of the pretraining methods once they have been fine-tuned on Ta-En parallel data: the results are very similar and all methods perform substantially better than the baseline, which is trained on parallel data only

[4] An alphabetic character is one belonging to the language in question: the Latin alphabet for English and the Tamil script, which is an abugida script.

[5] All models are learnt jointly over the languages used for training (English, Tamil and in one case German too). The vocabulary size is dependent on the model trained and is specified in the experimental details below.

but optimised in terms of training parameters and subword segmentation. We choose to use De-En pretraining for our final models and a mixture of De-En and mBART pretraining for intermediate MT models used for data augmentation (see the next paragraph).

Iterative backtranslation Data augmentation by backtranslating monolingual data has long been used in MT to provide greater amounts of in-domain parallel data in low resource settings (Bertoldi and Federico, 2009; Bojar and Tamchyna, 2011). We use backtranslation to translate the monolingual in-domain English and Tamil texts into the other language using an intermediate MT model and use the resulting synthetic parallel data to train new MT models.

We apply this iteratively (Hoang et al., 2018), as shown in Figure 1, to produce successively better MT models, initialising the models at each stage using either mBART or De-En pretraining. The intermediate MT models used to produce backtranslations are in white and the final models, which are then fine-tuned (as specified in the section entitled *Final model creation*) are in grey.

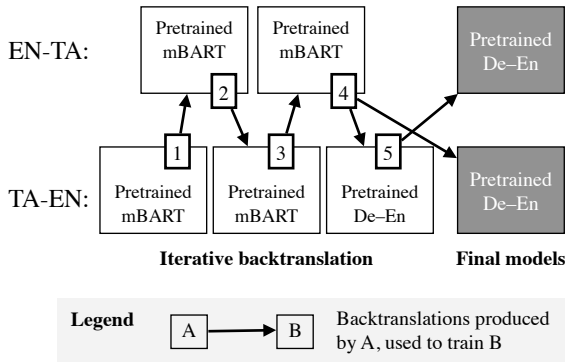


Figure 1: Iterative backtranslation process

1. We first train a Ta→En model initialised with mBART pretraining and fine-tuned on parallel data only. We then use this model to backtranslate all monolingual Tamil data into English.

2. We use the resulting backtranslated data together with the genuine parallel data to train an mBART-pretrained En→Ta model. After early stopping, we continue training using the genuine parallel data only. We then use this model to backtranslate 5M sentences of in-domain English data into Tamil.[6]

3. We use this new backtranslated data together with the genuine parallel data oversampled 7 times to train a second mBART-pretrained Ta→En model. After early stopping, we continue training using genuine parallel data only. We then use this model to backtranslate all the monolingual Tamil data.

4. We repeat step 2 with this latest backtranslated data, generating the final backtranslations to be used for the Ta→En direction.

5. We use 5M of these final backtranslations along with the Ta-En genuine parallel data oversampled 15 times to fine-tune a De-En pretrained model and use this to generate the final backtranslations to be used for the En→Ta direction.

The results of the iterative backtranslation steps on the dev set can be found in Table 4. They show increasing BLEU scores at each successive step.

System	Pretraining	BT dataset	EN→TA dev	EN→TA test	TA→EN dev	TA→EN test
1	mBART	-	-	-	14.00	13.40
2	mBART	1	10.50	5.68	-	-
3	mBART	2	-	-	18.60	**15.19**
4	mBART	3	**11.30**	**6.65**	-	-
5	De-En	4	-	-	**19.30**	-

Table 4: Results (BLEU scores) for the successive models used for backtranslations (BT) (as shown in Figure 1). Each row uses backtranslations produced by the system from the previous row.

System	EN→TA
Parallel-only baseline	5.10
(i) mBART pretraining	7.40
XLM BT	9.90
mBART BT	10.50
De-En BT	10.40
(ii) De-En pretraining	7.30
XLM BT	9.30

Table 5: Dev set results (BLEU scores) for alternative backtranslation schemes for system 2 from Figure 1.

In addition to the described strategy, we also experimented with training different pretrained models using backtranslations produced by different

[6]The En→Ta backtranslations at this step and the following steps are filtered using the same processing as described in Section 2.1, filtered using dual conditional cross-entropy filtering (Junczys-Dowmunt, 2018) and the top sentences are selected to train the next step.

models (e.g. training an mBart pretrained model on XLM-produced backtranslations). We report a small selection of these experiments here for one of the backtranslation steps, comparing the use of alternatives to system 2 (from Figure 1). These results are shown in Table 5: (i) a pretrained mBART model trained on backtranslations from each of the pretrained models, and (ii) a pretrained De-En model trained on XLM backtranslations. For this particular step of the iterative process, training a pretrained mBART model on backtranslations produced by the pretrained mBART model produced the best scores, explaining why this was chosen.

Final model creation Our final models are pretrained De-En models (in grey in Figure 1). After pretraining, the finalisation of these models follows a two-step training strategy to incorporate the synthetic and genuine parallel data:

1. We first train our models on the synthetic data described previously (20M sentences for Ta→En and just over 2.1M sentences for En→Ta).

2. We then fine-tune the models on a mixture of parallel and synthetic data.

This approach of pre-training on synthetic data and fine-tuning on genuine and synthetic data has been found to work well for other tasks (Junczys-Dowmunt and Grundkiewicz, 2016; Grundkiewicz et al., 2019). For the second step, we adopt different strategies for each language direction, depending on which worked best. For Ta→En, we fine-tune on genuine parallel data and 500k of the top scored backtranslations.[7] For En→Ta we fine-tune on a mixture of genuine parallel data, synthetic data produced using multi-agent dual learning (MADL; Wang et al., 2019; Kim et al., 2019) and the top 1M backtranslations. This MADL data comprises a mixture of forward translations and backtranslations of the parallel data created using intermediate models in both directions.

We also carried out preliminary experiments with multilingual training using other Indian languages and experiments with phrase-based MT using Moses (Koehn et al., 2007) but they did not achieve good results.

[7]Scoring is done using dual conditional cross-entropy filtering as specified in Footnote 6.

2.3 Experimental settings

We use the Marian toolkit (Junczys-Dowmunt et al., 2018) for all models except for those using XLM pretraining, for which we use the Facebook XLM toolkit (Lample and Conneau, 2019b). All models trained (including those used to produce backtranslations) use the Transformer-base architecture (Vaswani et al., 2017) with default hyperparameters according to the Marian or XLM implementation (6 encoder and 6 decoder layers, embedding dimension of 512, 16 heads, feedforward dimension of 2,048, standard learning rate warm-up).

Parallel-only baseline Our parallel-only baseline is trained with a joint vocabulary of size 5,414 for Ta-En and 418 for En-Ta. The En-Ta model was trained with a small batch size of 1000 tokens. mBart and XLM models are trained with a joint vocabulary size of 20,000 SentencePiece BPE subwords (including special tokens for language and domain tags, masking and sentence separators).

mBART training English and Tamil sentences are mixed in equal amounts in each batch. We use our re-implementation of mBART using Marian.[8] We deviate from the original implementation by always using two sentences per input segment, whereas the original paper used as many sentences as they could fit into the 512-token limit. The noise hyperparameters are the same as the original paper (35% of tokens are masked in contiguous spans of an average of 3.5 tokens. Masked spans do not cross sentence boundaries). Unlike XLM, we do not use online backtranslation during pre-training. We train until early stopping based on an held-out non-parallel dataset generated using the same noise function as the training data. During monolingual pretraining we early stop after the validation score (measured every 5,000 updates) does not improve for 10 consecutive times. When training on backtranslations or finetuning on parallel data we early stop on the parallel development corpus, measuring the valdation score every 500 updates.

De-En pretraining For models with De-En pretraining, we trained a SentencePiece model with a vocabulary size of 32k on roughly equal amounts of Tamil, English and German data (subsampling Ger-

[8]We implement an online "training harness" that reads monolingual sentences in English and Tamil, converts them to mBART training examples by applying noise and sends them to the Marian training process. Code and training scripts: https://github.com/Avmb/marian-mBART

man and English). The final MT vocabulary size is 49,213 as it is based on using all German-English data for training. The models are trained using tied target embeddings, a learning rate of 0.0002, the Adam optimiser (Kingma and Ba, 2015) and optimiser delay of 2 on 4 GPUs. We train all models until convergence based on the BLEU score on the held-out dev set provided for the task.

2.4 Results

Table 6 shows the final automatic evaluation score of our submissions for both directions on the dev set and the test set, including an ablation of the various components: pretraining using the De-En MT data (and fine-tuned on parallel data), addition of synthetic data to this setup and finally fine-tuning of the resulting model as specified previously.

System	EN→TA		TA→EN	
	dev	test	dev	test
Parallel-only baseline	5.10	3.10	10.10	10.60
Our final models				
Pretraining (De-En)	7.30	5.00	13.60	14.20
+ synthetic data	11.90	7.90	18.80	12.60
+ fine-tuning	**12.30**	**8.40**	**21.00**	**16.60**

Table 6: EN↔TA results (BLEU scores) for the successive steps in the creatino of our final models. The Last row represents the primary submission systems.

The best results (8.40 for En-Ta and 16.60 for Ta-En) are achieved with all three approaches to training. We found that ensembling did not improve our results and therefore our submitted systems are single models. We note that our final approach sees a big difference in the BLEU score between the dev and test sets. While BLEU scores are not directly comparable across datasets, the drop is quite significant and could indicate a domain shift between the two sets. Our models rely heavily on the use of backtranslated data and therefore could be adapting to translationese, which is rewarded in the dev set but not in the test set.

3 English↔Inuktitut

Compared to English-Tamil, the English-Inuktitut language pair is relatively well-resourced at approximately 1.3M sentence pairs. As such we were able to train conventional bilingual Transformer systems, which formed the basis of our submission. We also trained multilingual systems, but opted not to use these in our submission as results on the dev set did not appear to be promising (although evaluation proved challenging for this pair due to overlap between the training and dev data). Post-submission evaluation showed that our multilingual systems actually outperformed our submitted systems on the test sets.

3.1 Data and Preprocessing

We used all of the Nunavut Hansard data provided by the task organisers. For Inuktitut→English, this was supplemented with a similar volume of synthetic data, back-translated from the English side of the Europarl and News Crawl corpora. The only additional monolingual Inuktitut data was 163k sentences of common-crawl data, which we back-translated for the English→Inuktitut system.

We developed two multilingual systems: English→ {Inuktitut,German,Russian} and {Inuktitut,German,Russian} → English. The Russian and German languages were selected due to the availability of suitable volumes of data in the domains of interest (news and parliamentary proceedings). Both multilingual systems used the same dataset, which contains genuine iu-en, synthetic iu-en, genuine de-en, and genuine ru-en in a ratio of approximately 1:1:2:2 (both systems used all of the synthetic data, regardless of back-translation direction). Table 7 lists all of the corpora used for the multilingual systems

Lang. Pair	Size	Corpus
en-iu	1,310k	Nunavut Hansard
en-iu	650k	Synthetic (from en Europarl)
en-iu	650k	Synthetic (from en News 2019)
en-iu	163k	Synthetic (from iu CommonCrawl)
en-de	361k	News Commentary
en-de	1,817k	Europarl
en-de	400k	Paracrawl
en-ru	1,000k	Yandex
en-ru	1,600k	UN

Table 7: Data used for the multilingual English-Inuktitut models. Size is given in sentence pairs.

For the bilingual systems, our preprocessing pipeline consisted of corpus cleaning and segmentation. For corpus cleaning, we used the `clean-corpus-n.perl` script from the Moses toolkit (Koehn et al., 2007). This applies a maximum length threshold of 80 as well as removing empty sentences and sentence pairs with length ratios greater than 9:1.

For segmentation, we trained language-specific SentencePiece models (Kudo and Richardson, 2018) with a vocabulary size of 32,000 BPE subwords and a vocabulary threshold of 50.

Preprocessing was identical for the multilingual systems except that for the English→ {de,iu,ru} we added a token to each source sentence to specify the target language (as in Johnson et al. (2017)).

After the release of the test set, the task organisers reported that some test sentences had been enclosed in extraneous quotes. For our submission, and for post-submission evaluation, we removed outer quotes prior to translation for any test sentence that began and ended with a double quote character.

3.2 Experimental Settings

We used the Nematus toolkit (Sennrich et al., 2017) for all models. For preliminary systems, our hyperparameter settings matched the 'base' configuration of Vaswani et al. (2017). We used these systems for back-translation. For the multilingual systems and final bilingual systems, our settings matched Vaswani et al. (2017)'s 'big' configuration. We used a batch size of 16,384 tokens for all models.

Since the bilingual 'big' systems looked the most promising during development, we trained a second model for each direction and used ensembling in our submission systems.

3.3 Results

Table 8 shows the automatic evaluation scores for our submitted ensemble systems as well as individual bilingual systems and multilingual systems.

Post-submission evaluation on the test set shows that the multilingual systems outperformed the bilingual systems, which is in contrast to the results obtained on the dev sets during system development. We suspect that the large differences in BLEU between dev/test and bilingual/multilingual to overlap between the Nunavut training and dev data. We found that a large proportion of dev sentences were present in the training data, although many were short, frequently used phrases, such as 'Thank you, Mr. Speaker.' and 'The motion is carried.' During development we tried filtering the dev set to reduce overlap at the sentence level. This lowered the scores, but still produced the same overall order: bilingual big > bilingual base > multilingual and so we used this result to guide our decision on which systems to submit. With hindsight, we suspect that the prevalence of formulaic, but not necessarily identical, constructions in the text may be a complicating factor and that more aggressive filtering of the dev set may have produced more robust results. Compared to the bilingual base or multilingual models, the bilingual big models have more capacity available for memorisation of the training data and it seems that our filtering was not enough to counter this effect.

4 Conclusion

In this submission we focused on a low-resource language pair (English-Tamil) and a medium-resource language pair (English-Inuktitut). All our translation systems are based on the Transformer architecture. We found it beneficial to use monolingual data in the form of backtranslations. In the case of En-Ta, we saw notable gains by using pretraining using both the denoising autoencoding (mBART) objective and multilinguality in the form of German-English pretraining. However, we were not able to gain any quality from multilingual training on data for other Indian languages. For English-Inuktit, multilinguality did not appear to help on the dev set, but was found, post-submission, to help on the test set.

In general, we found that English-Tamil is a much more challenging task, where pretraining is absolutely necessary to reach acceptable quality, while for English-Inuktit reasonable translation quality can be achieved using only parallel data.

Acknowledgements

This work was supported by funding from the European Union's Horizon 2020 research and innovation programme under grant agreements No 825299 (GoURMET), 825303 and the UK Engineering and Physical Sciences Research Council (EPSRC) fellowship grant EP/S001271/1 (MT-Stretch). The research presented in this publication was conducted in cooperation with Samsung Electronics Polska sp. z o.o. - Samsung RD Institute Poland.

References

Alham Fikri Aji, Nikolay Bogoychev, Kenneth Heafield, and Rico Sennrich. 2020. In neural machine translation, what does transfer learning transfer? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7701–7710, Online.

Rachel Bawden, Nikolay Bogoychev, Ulrich Germann, Roman Grundkiewicz, Faheem Kirefu, Antonio Valerio Miceli Barone, and Alexandra Birch. 2019. The university of Edinburgh's submissions to the WMT19 news translation task. In *Proceedings of*

System	EN→IU			IU→EN		
	dev	dev-filt	test	dev	dev-filt	test
Transformer base	22.0	9.8	7.7	35.9	24.3	22.4
Transformer big (1)	24.9	11.4	8.0	45.4	28.3	21.6
Transformer big (2)	24.8	11.5	7.9	45.8	28.4	21.7
Ensemble of (1) and (2)	27.0	12.9	8.2	48.8	31.0	23.0
Multilingual	16.5	8.0	9.7	34.8	24.5	23.3

Table 8: EN↔IU automatic evaluation results (BLEU) on the WMT20 dev and test sets. We also include results for our filtered version of the dev set.

the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1), pages 103–115, Florence, Italy.

Nicola Bertoldi and Marcello Federico. 2009. Domain adaptation for statistical machine translation with monolingual resources. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 182–189, Athens, Greece.

Ondřej Bojar and Aleš Tamchyna. 2011. Improving translation model by monolingual data. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 330–336, Edinburgh, UK.

Isaac Caswell, Ciprian Chelba, and David Grangier. 2019. Tagged back-translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 53–63, Florence, Italy.

Roman Grundkiewicz, Marcin Junczys-Dowmunt, and Kenneth Heafield. 2019. Neural grammatical error correction systems with unsupervised pre-training on synthetic data. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 252–263, Florence, Italy.

Vu Cong Duy Hoang, Philipp Koehn, Gholamreza Haffari, and Trevor Cohn. 2018. Iterative back-translation for neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 18–24.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Marcin Junczys-Dowmunt. 2018. Dual conditional cross-entropy filtering of noisy parallel corpora. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 888–895, Belgium, Brussels.

Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2016. Log-linear combinations of monolingual and bilingual neural machine translation models for automatic post-editing. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 751–758, Berlin, Germany.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia.

Young Jin Kim, Marcin Junczys-Dowmunt, Hany Hassan, Alham Fikri Aji, Kenneth Heafield, Roman Grundkiewicz, and Nikolay Bogoychev. 2019. From research to production and back: Ludicrously fast neural machine translation. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 280–288, Hong Kong.

Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations*, ICLR'15, San Diego, California, USA.

Tom Kocmi and Ondřej Bojar. 2018. Trivial transfer learning for low-resource neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 244–252, Brussels, Belgium.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium.

Guillaume Lample and Alexis Conneau. 2019a. Cross-lingual language model pretraining. *Advances in Neural Information Processing Systems (NeurIPS)*.

Guillaume Lample and Alexis Conneau. 2019b. Cross-lingual Language Model Pretraining. In *arXiv:1901.07291*.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium.

Rico Sennrich, Orhan Firat, Kyunghyun Cho, Alexandra Birch, Barry Haddow, Julian Hitschler, Marcin Junczys-Dowmunt, Samuel Läubli, Valerio Antonio Miceli Barone, Jozef Mokry, and Maria Nadejde. 2017. Nematus: a toolkit for neural machine translation. In *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 65–68.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Yiren Wang, Yingce Xia, Tianyu He, Fei Tian, Tao Qin, ChengXiang Zhai, and Tie-Yan Liu. 2019. Multi-agent dual learning. In *International Conference on Learning Representations*.

GTCOM Neural Machine Translation Systems for WMT20

Chao Bei, Hao Zong, Qinming Liu and Conghu Yuan
Global Tone Communication Technology Co., Ltd.
{beichao, zonghao, liuqingmin and yuanconghu}@gtcom.com.cn

Abstract

This paper describes the Global Tone Communication Co., Ltd.'s submission of the WMT20 shared news translation task. We participate in four directions: English to (Khmer and Pashto) and (Khmer and Pashto) to English. Further, we get the best BLEU scores in the directions of English to Pashto, Pashto to English and Khmer to English (13.1, 23.1 and 25.5 respectively) among all the participants. Our submitted systems are unconstrained and focus on mBART (Multilingual Bidirectional and Auto-Regressive Transformers), back-translation and forward-translation. Also, we apply rules, language model and RoBERTa model to filter monolingual, parallel sentences and synthetic sentences. Besides, we validate the difference of the vocabulary built from monolingual data and parallel data.

1 Introduction

We participated in the WMT shared news translation task and focus on the bidirections: English and Khmer, English and Pashto. We applied fairseq(Ott et al., 2019) as our develop tool and use transformer(Vaswani et al., 2017) as the main architecture. The primary ranking index for submitted systems is BLEU (Papineni et al., 2002), therefore we apply BLEU as the evaluation matrix for our translation system. For Khmer, we use polyglot [1] as the tokenizer before evaluation.

For data preprocessing, the basic method includes punctuation normalization for all language. Further, according to the different language characteristics. Tokenization, truecase and byte pair encoding (BPE) (Sennrich et al., 2015b) are applied for English, and sentencepiece (Kudo and Richardson, 2018) is applied for Khmer and Pashto. Besides, human rules, language model and RoBERTa model (Liu et al., 2019) are also involved to clean parallel data, monolingual data and synthetic data. Regard to the techniques on model training, back-translation (Sennrich et al., 2015a) and forward-translation are applied to verify whether these techniques could improve the translation performance especially in low-resource condition.

We all know that it is more difficult to train a model in low-resource condition, because it suffers from data sparsity and out-of-vocabulary problem. Normally knowledge distillation (Kim and Rush, 2016) is a good way to generate synthetic data. But in this task we suppose that knowledge distillation can only generate 100 thousand to 1 million parallel sentences due to the size of provided data. Therefore, we use forward-translation with monolingual data to generate more synthetic data. Here forword-translation refers to translate the source sentences to target language, and clean synthetic data.

This paper is arranged as follows. We firstly describe the task and show the data information, then introduce how we do data filtering, including human rules, language model and RoBERTa model. After that, we describe the techniques on low-resource condition and show the conducted experiments in detail of all directions, including data preprocessing, model architecture, back-translation and forwor-translation. At last, we analyze the results of experiments and draw the conclusion.

2 Task Description

The task focuses on bilingual text translation in news domain and the provided data is show in Table 1, including parallel data and monolingual data. For the direction between English and Khmer, the parallel data is mainly from ParaCrawl v5.1 and shared task on parallel corpus filtering (mostly from OPUS (Tiedemann, 2012)), as well as the direction between English and Pashto. Another, monolin-

[1] https://github.com/aboSamoor/polyglot

Proceedings of the 5th Conference on Machine Translation (WMT), pages 100–104

language	number of sentences
en-ps parallel data	1M
en-km parallel data	4.17M
en monolingual data	16.9M
ps monolingual data	4.2M
km monolingual data	12.7M
en-ps development set	3162
en-km development set	2378
en-ps devtest set	2698
en-km devtest set	2309

Table 1: Task Description.

configuration	value
architecture	transformer
word embedding	512
Encoder depth	5
Decoder depth	5
transformer heads	2
size of FFN	2048
attention dropout	0.2
dropout	0.4
relu dropout	0.2

Table 2: The FLoRes model architecture.

configuration	value
architecture	transformer
word embedding	768
Encoder depth	6
Decoder depth	6
transformer heads	12
size of FFN	3072
attention dropout	0.1
dropout	0.1
relu dropout	0

Table 3: The mBART model architecture.

gual data we used are News crawl both for English, Common Crawl and Wiki dumps both for Khmer and Pashto. All directions we participated are new for this year, we use wikipediadev as our development set and wikipediadevtest as our test set.

3 Data Filtering

The methods of data filtering are mainly the same as we did in last year (Bei et al., 2019), including human rules and language model. Further, another methods we used this year are as follows:

Clean repeated translation sentences in synthetic data. For example, we often see the translation like: I want to eat an apple apple apple apple apple, when translating a source language sentence with repeated words until the end of the sentence. In this task, we made a simple clean strategy which is to remove the sentences that repeat one word four times, two words three times or three words two times.

Clean synthetic data by RoBERTa model. In order to clean synthetic data, especially from forword-translation, we represent the source and target sentences by RoBERTa model and calculate the cosine distance. Remove the sentences with low score or without translation (the source sentence and target sentence are same).

4 Forward-translation

In low-resource condition, out-of-vocabulary is a problem. There is a difference between the test scenario and training scenario, which means the words appear in test set may be not existed in training vocabulary. Back-translation is a common way to extend the word vocabulary. However, with the generated synthetic data from back-translation, only target vocabulary can be enriched. To extend the source side vocabulary, we use source-to-target model to translate the source monolingual data to target side. Further, it is necessary to clean the forward-translation sentences to avoid cascading error for the next training. We use RoBERTa to represent the source and target sentence and calculate the cosine distance. Remove the sentences with low score or without translation (the source sentence and target sentence are same).

5 Experiment

5.1 Model architecture

- **Baseline** Table 2 shows the baseline model architecture.
- **mBART** We fine-tune on mBart model to get better translation. Table 3 shows the model architecture.
- **Big transformer** We use transformer big model to train our model with fairseq. The model configuration and training parameters is almost same as last year we use. In order to training more stable in low-resource condition, we add layer normalize before encoder and decoder.

5.2 Training Step

This section introduces all the experiments we set step by step and Figure 1 shows the whole flow.

- **Date Filtering** Following the task of Parallel Corpus Filtering and Alignment for Low-Resource Conditions, we use the LASER-based scores to filter the raw parallel sentences and extract 5 million words England tokens.
- **Baseline.** We use FLoRes (Guzmán et al., 2019) architecture to construct our baseline in low-resource condition.
- **Fine-tuning on mBART.** In such low-resource condition, we fine-tune on mBART model with filtered sentences.
- **Back-translation.** We use fine-tuned model to translate the target sentence to source side, and clean synthetic data with language model and RoBERTa model. Mix cleaned back-translation data and parallel sentences and fine-tune on mBART model.
- **Forward-translation.** Source side sentences are translated to target side, and cleaned by language model and RoBERTa model. Mixed with cleaned back-translation data, forward-translation data and parallel sentences, fine-tune on mBART model.
- **Monolingual vocabulary.** To enrich the vocabulary further, we preprocess the monolingual data and build the vocabulary as model vocabulary. Here, we normalize the punctuation of all data by nomalize-puncuation.perl in Moses toolkit (Koehn et al., 2007). We apply tokenizer and truecaser in Moses toolkit for English. Finally, BPE (Byte Pair Encoding) (Sennrich et al., 2016) is applied on tokenized English and sentencepiece is applied on Pashto and Khmer. The BPE and sentencepiece merge operation are both 32000. Therefore, the vocabulary of monolingual data is set to 32500. We use these vocabularies as model vocabulary and train big transformer model.
- **Joint training.** Repeat back-translation step and forward-translation step by best model, until there is no improvement.
- **Ensemble Decoding.** We use GMSE Algorithm (Deng et al., 2018) to select models to obtain the best performance.

6 Result and analysis

Table 4 and Table 5 show the BLEU score we evaluated on development set for English to Pashto, Pashto to English, English to Khmer and Khmer to English respectively.

For fine-tuning on mBART model, we find that it is the most effective method with an improvement from 0.56 to 3.62 BLEU score in low-resource condition. And back-translation gets the improvement from 0.17 to 3.04 BLEU score. Forward translation and monolingual vocabulary enrich the information in low-resource condition, with improvement of 0.16 to 0.74 BLEU score and 0.69 to 0.94 BLEU score respectively. Further, joint training and ensemble decoding slightly increase the performance with 0.31 to 0.4 BLEU score and 0.15 to 0.4 BLEU score.

7 Summary

This paper describes GTCOM's neural machine translation systems for the WMT20 shared news translation task. For all translation directions, we build systems mainly base on mBART model and enrich information by back-translation, forward-translation and using monolingual vocabulary with data filtering, including calculating cosin distance by RoBERTa model, language model and so on. The effect of increasing information is also dependent on data filtering. Finally, we submit the online system including English to Pashto, Pashto to English, Khmer to English and English to Khmer with almost same methods in this paper. Another, we also submit our online system from English to Tamil and Tamil to English.

Acknowledgments

This work is supported by 2020 Cognitive Intelligence Research Institute[2] of Global Tone Communication Technology Co., Ltd.[3]

References

Chao Bei, Hao Zong, Conghu Yuan, Qingming Liu, and Baoyong Fan. 2019. GTCOM neural machine translation systems for WMT19. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 116–121, Florence, Italy. Association for Computational Linguistics.

[2]http://www.2020nlp.com/
[3]http://www.gtcom.com.cn/

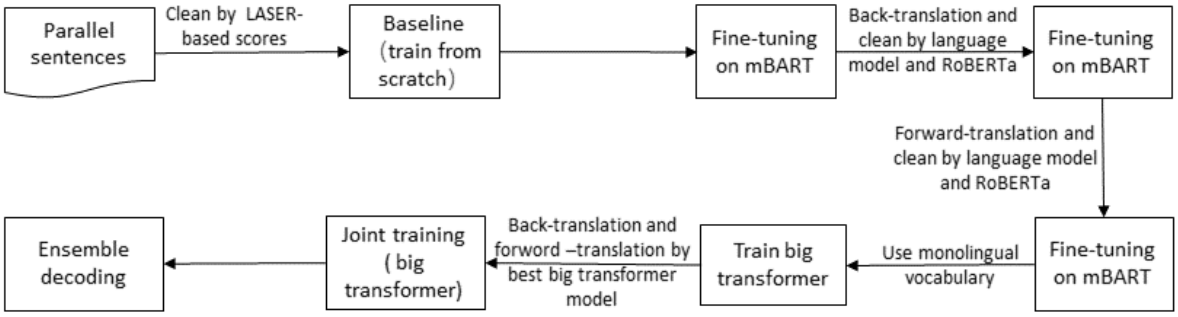


Figure 1: The whole work flow.

model	en2ps	ps2en
baseline	5.73	8.37
fine-tuning on mBART	9.35	11.98
+ back-translation	9.97	12.15
+ forward-translation	10.15	12.31
+ monolingual vocabulary	10.88	13.25
+ joint training	11.19	13.69
+ Ensemble Decoding	11.34	14.09

Table 4: The case-sensitive BLEU score between English and Pashto.

model	en2km	km2en
baseline	8.68	7.47
fine-tuning on mBART	9.24	9.91
+ back-translation	12.28	12.23
+ forward-translation	12.85	12.97
+ monolingual vocabulary	13.54	13.73
+ joint training	13.87	14.04
+ Ensemble Decoding	14.13	14.29

Table 5: The case-sensitive BLEU score between English and Khmer.

Yongchao Deng, Shanbo Cheng, Jun Lu, Kai Song, Jingang Wang, Shenglan Wu, Liang Yao, Guchun Zhang, Haibo Zhang, Pei Zhang, et al. 2018. Alibaba's neural machine translation systems for wmt18. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 368–376.

Francisco Guzmán, Peng-Jen Chen, Myle Ott, Juan Pino, Guillaume Lample, Philipp Koehn, Vishrav Chaudhary, and Marc'Aurelio Ranzato. 2019. The flores evaluation datasets for low-resource machine translation: Nepali-english and sinhala-english. *arXiv preprint arXiv:1902.01382*.

Yoon Kim and Alexander M Rush. 2016. Sequence-level knowledge distillation. *arXiv preprint arXiv:1606.07947*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*, pages 177–180. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible

toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015a. Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015b. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *Lrec*, volume 2012, pages 2214–2218.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

DiDi's Machine Translation System for WMT2020

Tanfang Chen, Weiwei Wang, Wenyang Wei,
Xing Shi, Xiangang Li, Jieping Ye, Kevin Knight
AI Labs, DidiChuxing
{chentanfang, wangweiweiwill, weiwenyang,
xingshi, lixiangang, yejieping, kevinknight}@didiglobal.com

Abstract

This paper describes DiDi AI Labs' submission to the WMT2020 news translation shared task. We participate in the translation direction of Chinese→English. In this direction, we use the Transformer as our baseline model, and integrate several techniques for model enhancement, including data filtering, data selection, back-translation, fine-tuning, model ensembling, and re-ranking. As a result, our submission achieves a BLEU score of 36.6 in Chinese→English.

1 Introduction

We participate in the WMT2020 news translation shared tasks in Chinese → English direction. For this translation direction, we train several variants of Transformer (Vaswani et al., 2017) models on the provided parallel data enlarged with synthetic data from monolingual data. We experiment with several techniques proposed in the past translation tasks and adopt effective ones as components of our system.

Our data preparation pipeline consists of data filtering, data augmentation, and data selection. For data filtering, we filter sentence pairs based on language model scoring, alignment model scoring, etc. For data augmentation, we experiment with iterative back-translation (Sennrich et al., 2016; Edunov et al., 2018) methods and iterative knowledge distillation (Freitag et al., 2017) methods. We leverage source-side monolingual data by applying iterative knowledge distillation, and target-side monolingual data by back-translation methods, including greedy search, beam search, and noised beam search. For data selection, we select an in-domain corpus with N-grams language models and binary classifiers. A tri-gram token-level language model and a bi-gram character-level language model are introduced for English and Chinese respectively. Out-of-domain sentences which have similar scores as in-domain sentences are chosen. We also treat data selection as a text classification problem, and use BERT (Devlin et al., 2019) as the basic classifier. In this way, we collect a corpus of high-quality in-domain training data, which improves translation performance significantly.

To enhance a single model, we use several variants of Transformer, including Transformer with relative position attention (Shaw et al., 2018), Transformer with larger feedforward inner (FFN) size ($8,192$ or $15,000$), and Transformer with reversed source. We then ensemble these models with adequate model diversity and data diversity to further improve the performance.

Domain conflicts influence the translation performance significantly. For example, there exist differences between written English and spoken English. Usually, a model cannot do the best in all domains due to the conflicts. In this work, we propose to obtain domain information with unsupervised clustering and exploit this information for translation. Specifically, we partition the training data, dev data, and test data into different clusters, and translate each cluster part of the test set with the model fine-tuned on the corresponding training set. Exploiting domain information helps improve the translation significantly. Details will be discussed in Section 3.

This paper is structured as follows: Section 2 describes variants of Transformer we used in the competition. In Section 3, we introduce several techniques for model enhancement, including data filtering, back-translation, fine-tuning, model ensembling. Section 4 presents experimental settings, results and analysis. Finally, in Section 5 we draw a brief conclusion of our work in the WMT2020.

Proceedings of the 5th Conference on Machine Translation (WMT), pages 105–112
Online, November 19–20, 2020.

2 Model

2.1 Transformer

The Transformer adopts a sequence-to-sequence structure, using stacked encoder and decoder layers of self-attention. Encoder layers consist of a self-attention layer followed by a feed-forward layer. Decoder layers consist of a masked self-attention layer, an encoder-decoder attention layer, and a feed-forward layer to incorporate source information and generate texts. The residual connections (He et al., 2016) and layer normalization (Ba et al., 2016) are introduced in the encoder and decoder layers for better convergence. In contrast to recurrent neural networks, the Transformer implicitly leverages relative and absolute position information in its structure. The Transformer introduces position encoding based on sinusoids in its inputs to incorporate position information.

In the competition we use Transformer Big as the baseline model, in which both the encoder and decoder have 6 layers, the number of heads is 16, the hidden size is $1,024$, and the feedforward inner (FFN) size is $4,096$.

2.2 Transformer with Relative Position Attention

The original Transformer leverages position information by taking absolute positional embeddings as inputs and does not explicitly capture the information in its structure. Thus the original Transformer cannot leverage position information efficiently. Here we used relative positional embeddings in the self-attention mechanism proposed in Shaw et al. (2018) for the encoder layers and decoder layers. We do an ablation study and find that the model with relative positional embedding has faster convergence and better performance than Transformer Big. We adopt Transformer with relative position attention as a basic architecture in the final ensemble model.

2.3 Transformer with Larger FFN Size

Since increasing the model size can help improve the performance on the NMT tasks, we experiment with Transformer with a larger embedding dimension, FFN size, number of heads, and number of layers. We find that using a larger FFN size ($8,192$ or $15,000$) gives a reasonable improvement in the performance while maintaining a manageable network size. We adopt a Transformer with FFN size of $8,192$ and a Transformer with FFN size of $15,000$ as basic models in the final ensemble model, which has a larger inner dimension of feed-forward network than Transformer Big. Since Transformer with a larger FFN size is more likely to overfit, we set the dropout rate from 0.1 to 0.3 and use a label smoothing rate of 0.2.

2.4 Transformer with Reversed Source

We reverse the source sentences of the bilingual corpus and train a Transformer with source reversed. In this way, the model can learns a different meaning of the positional embeddings, which helps capture the source sentences from a different perspective. Viewing source in a reversed order provides another kind of model diversity and data diversity and presents positive effects in the final model ensemble.

3 System Overiew

3.1 Data Filtering

Previous works (Sun et al., 2019; Xia et al., 2019; Guo et al., 2019) show that the translation performance improves as the quality of parallel corpus improves. We filter the training bilingual corpus with the following schemes:

- Normalize punctuation with Moses scripts
- Filter out the sentences longer than 120 words or sentences including a single word more than 40 characters.
- Filter out the sentences which contain HTML tags or duplicated translations.
- Filter out the sentences whose languages detected by fastText[1] (Joulin et al., 2017) are not identical to the translation direction.
- Filter out the sentences whose alignment scores obtained by fast-align[2] (Dyer et al., 2013) are low.
- Filter out the sentences whose n-gram scores from language models are low.
- Filter out the sentences whose length ratio between the source and target are not in range of $1:3$ and $3:1$

[1] https://github.com/facebookresearch/fastText
[2] https://github.com/clab/fast_align

In this paper, we also filter out noisy sentence pairs with the translation acceptability filter proposed in (Zhang et al., 2020). Specifically, we feed the sentence pair (s, t) into multilingual BERT, which accepts two-sentence input due to its next-sentence prediction objective. Instead of using the [CLS] token representation, we use a Convolutional Neural Network (CNN) layer that takes the BERT output and generates the final representation of the pair. Our experiments show that using CNN layer pooling achieves marginal gains over [CLS] pooling. We use the softmax probability as the degree of parallelism and filter the sentences. The translation quality of the model boosts with the data filtering strategies.

3.2 Large-scale Back-Translation

The provided monolingual data contains a certain amount of noise, in which noise may affect the translation quality implicitly. Therefore, we adopt the data filtering schemes described in Section 3.1.

Previous work (Edunov et al., 2018) shows that leveraging the back-translation mechanism on the large-scale monolingual corpus can help improve the translation quality. Edunov et al. (2018) investigates several methods to generate synthetic source sentences, including greedy search, beam search, sampling top-K outputs, adding noise to beam search output, and adding noise to input sentences.

- Both greedy search and beam search are approximate algorithms to identify the maximum a-posteriori (MAP) output, i.e. the sentence candidate with the largest estimated probability given an input. This leads to less rich translations and is particularly problematic for text generation tasks such as back-translation.

- Sampling top-K method selects the k most likely tokens from the output distribution, renormalizes, and samples from this restricted set. This method is a trade-off between MAP and unrestricted sampling.

- Adding noise to input sentences or beam search outputs can help improve the quality and robustness of the translation.

We experiment with the above methods and observe that language pairs with abundant parallel corpus like Chinese → English obtain obvious improvement with beam search and adding noise. In our back-translation scheme, we add noise to input sentences, and use a beam search to produce the synthetic sentences. In particular, we delete words, replace words by a filler token and swap words according to a random permutation with the probability of 0.05.

Zhang et al. (2018) proposed an iterative joint training of the source-to-target model and target-to-source model for the better quality of synthetic data. Specifically, in each iteration, the target-to-source model is responsible for generating synthetic parallel training data for the source-to-target model using the target-side monolingual data. At the same time, the source-to-target model is employed for generating synthetic bilingual training data for the target-to-source model using the source-side monolingual data. The performance of both the target-to-source and source-to-target model can be further improved iteratively. We stop the iteration when we can not achieve further improvement.

Since there are amounts of genres in both parallel and synthetic data, we adopt a language model to divide data into a coarse domain-specific corpus. We train multiple language models on different types of monolingual data (News crawl, Gigaword, etc.), and score the sentences with the language models. We select the top 600K sentences for each domain. In the final submission, we adopt an iterative joint training scheme and train models on both bilingual and synthetic data of different genres to improve translation quality.

3.3 Knowledge Distillation

Alternate knowledge distillation (Hinton et al., 2015; Freitag et al., 2017) and ensemble iteratively is adopted in the competition to further boost the performance of a single model. We simply use an ensemble model as the teacher model and boost the single student model by data augmentation. In our experiments, we use Transformer Big, Transformer with relative position, Transformer with larger FFN size, and Transformer with reversed source as basic models. For each model type, we ensemble other model types as the teacher model to boost the model performance. For example, the ensemble model of a Transformer with relative position, a Transformer with larger FFN size, and a Transformer with reversed source are adopted as a teacher model to improve the performance of a

Transformer Big.

Considering that distillation from a poor-quality teacher model is likely to hurt the student network and thus results in an inferior performance, we selectively use distillation in the training process. In our experiments, we filter out data according to the sentence-level BLEU scores whose English translations lower than 28.

3.4 In-domain Data Selection and Fine-tuning

Domain adaptation plays an important role in improving the performance towards given test data. A practical method for domain adaptation is training on the large-scale data and then fine-tuning on the in-domain data (Luong and Manning, 2015). We select the small in-domain corpus with several approaches, including N-grams language model similarity and binary classification.

N-grams: We adopt the algorithm proposed in Duh et al. (2013); Axelrod et al. (2011), which selects sentence pairs from the large out-of-domain corpus that are similar to the in-domain data. In our work, we train a tri-grams token-level language model for English and a bi-grams character-level language model for Chinese. We use the parallel texts as the out-of-domain corpus and all available test sets in the past WMT tasks and News Commentary as the in-domain corpus. We score the sentence pairs with bilingual cross-entropy differences as follows:

$$CE(H_{I-SRC}, H_{O-SRC}) + CE(H_{I-TGT}, H_{O-TGT}) \quad (1)$$

where we denote out-of-domain corpus as O, in-domain corpus as I. H_{I-SRC} denotes language models over the source side and H_{I-TGT} denotes language models over the target side on in-domain data. H_{O-SRC} denotes language models over the source side and H_{O-TGT} denotes language models over the target side on out-of-domain data. CE denotes the cross-entropy function which evaluates the differences between distributions.

Finally, we sort all sentence pairs and select the top 600K sentences with the lowest scores to fine-tuning the parameter of the model.

Binary Classification: We also treat in-domain data selection as a text categorization problem. There are two categories: in-domain (1) and out-of-domain (0). We use the pre-trained language model BERT as the basic classifier. For the fine-tuning data, all available newstest data and News Commentary are regarded as positive data, and randomly sampled data from the large-scale corpus are regarded as negative data. Then BERT is exploited to score the sentence pairs. We sort all sentence pairs and select the top 600K sentences with the highest scores as fine-tuning data.

All the in-domain data obtained by the above methods are adopted to fine-tuning the single model and provide about a 2 BLEU scores improvement.

3.5 Model Ensemble

Ensemble learning is a widely used technique in the real-world tasks, which provides performance improvement by taking advantages of multiple single models. In neural machine translation, a practical way of the model ensemble is to combine the full probability distribution over the target vocabulary of different models at each step during sequence prediction. We experiment with the max, avg, and log-avg strategies, and find the log-avg strategy achieves the best performance. We implement a model ensemble module in OpenNMT[3] (Klein et al., 2017). In our experiments, we observe that simply enlarging the size of ensemble models does not necessarily improve translation performance. However, brute-force search of all models is prohibitively expensive and unrealistic. As the number of models increases, the decoding of the ensemble will take more time than a single model and exceed the limits of computer resource capacity. Therefore, we adopt a greedy model ensemble algorithm (Li et al., 2019) as shown in Algorithm 1.

Since model and data diversity are important factors for an ensemble system, we train diverse models with different initialization seeds, different parameters, different architectures, and different training data sets. All the models are fine-tuned to achieve superior performance.

3.6 Domain Style Translation

Translation performance differs in different topic domains. For intuitive explanation, we take native style and translation style as an example, and our topic domains are generated by using unsupervised clustering, not limited to these two styles. Native style and translation style are much different. A single model cannot do the best in both styles. For the Chinese $\rightarrow$ English task in WMT 2017 and 2018, the source side of both dev set and test set

[3]https://github.com/OpenNMT/OpenNMT-tf

Algorithm 1: An simple ensemble algorithm based on greedy search

Input: a model list Ω_{cand} sorted by the scores on development data.
Output: a final model list Φ_{final}

```
for all combination of 2 models that model ∈ top-8 models do
    obtain translation by ensemble decoding and evaluate with BLEU score;
end
Choose the best 2 model combination as the initial Φ_final;
while there is tiny improvement as the model number increases do
    choose one single model from the rest of Ω_cand to the Φ_final which performs better when combined with Φ_final;
end
```

are composed of two parts: documents created originally in Chinese (translation style) and documents created originally in English (native style). For the Chinese→English task, if the Chinese sentences are created from native Chinese corpus, then the corresponding English sentences are in translation style, so the model fine-tuned on these parallel sentences helps with translation style. Similarly, if the English sentences are created from native English corpus, the model fine-tuned on these sentences helps with native style. Previous work (Sun et al., 2019) shows exploiting translation style and native style achieves much better performance. In our work, we classify sentences into different topic categories (not limited to translation style and native style), and translate each specific part of the test set with the model fine-tuned on the corresponding training set.

Domain Label: We use pre-trained BERT models to extract [CLS] vector as the sentence embedding and obtain two clusters by K-Means clustering. We use the cluster id as the domain label.

Domain Classification: Pre-trained BERT models are fine-tuned as a text classification task, based on the source and target side with the domain label we defined above. In this way, we can select several fine-tuning data w.r.t. different topic domains.

Decoding Stage: Since the test data is composed of a mixed-genre data, we first classify the domain of each sentence in the test set and obtain the probabilities corresponding to each domain. Then we apply a weighted ensemble method to integrate NMT models. Specifically, when computing the output probability of the next word, we multiply the output probability in each domain-specific translation model with the corresponding domain probability of each sentence.

3.7 Re-ranking

We obtain n-best hypotheses with an ensemble model and then train a re-ranker using k-best MIRA (Cherry and Foster, 2012) on the validation set. K-best MIRA works with a batch tuning to learn a re-ranker for the n-best hypotheses. The features we use for re-ranking are:

- Length Features: length ratio and length difference between the source sentences and hypotheses
- NMT Features: scores from the ensemble model
- Language Model Features: scores from multiple n-gram language models

4 Experiments and Results

4.1 Experiment Setup

Our implementation of the Transformer models is based on the version 2.3.0 of OpenNMT-tf. We use Transformer Big as a basic model. Transformer Big has 6 layers in both encoder and decoder respectively, where each layer consists of a multi-head attention sublayer with 16 heads and a feed-forward sublayer with inner dimension 4096. The word embedding dimensions and the hidden state dimensions are set to 1024 for both encoder and decoder. In the training phase, the dropout rate $P_{dropout}$ is set to 0.1. Variants of Transformer described in Section 2 are adopted in the competition.

In the training phase, we use cross entropy as the loss function and apply label smoothing of 0.1. We use Adam (Kingma and Ba, 2014) as our optimizer, with parameters settings $\beta_1 = 0.9$, $\beta_2 = 0.98$ and $\epsilon = 10^{-8}$. The initial learning rate is set to 10^{-4} for training and 10^{-5} for fine-tuning. The models are trained on 4 GPUs for about $500,000$ steps. Each model learns from data randomly sampled from the whole corpus, including bilingual data, synthetic data from back-translation, and synthetic data from knowledge distillation. Models

	Transformer Big	Transformer with relative position attention	Transformer with larger FFN size	Transformer with reversed source
baseline	26.01	26.23	26.12	26.08
+ data augmentation	27.02	27.03	27.13	26.69
+ In-domain data finetuning	29.33	29.49	29.62	29.18
+ model ensemble	29.72			
+ domain style weighted	31.77			
+ reranking*	**31.86**			

Table 1: BLEU evaluation results on the WMT 2018 Chinese → English test set (* denotes the submitted system)

	newstest19
baseline	26.19
+ data augmentation	27.45
+ In-domain data finetuning	37.23
+ model ensemble	37.64
+ domain style weighted	38.59
+ reranking	**38.99**

Table 2: BLEU evaluation results on the WMT 2019 Chinese → English test set

	newstest18	newstest19
NEU (Li et al., 2019)	30.9	34.2
MSRA (Xia et al., 2019)	30.9	**39.3**
Baidu (Sun et al., 2019)	31.83	38
ours	**31.86**	38.99

Table 3: Comparison with related work on the WMT 2018 and 2019 Chinese → English test set

used in iterative back-translation and knowledge distillation are trained for $200,000$ steps. We validate the model every $1,000$ steps on the development data and save the checkpoints with the best BLEU scores. After training, we average the last 10 checkpoints for every single model of the general domain.

In the fine-tuning phase, we use the averaged model obtained in the training phase as pre-train weights for domain models, and train with in-domain data selected as in Section 3.4 for $10,000$ steps without early stop. After fine-tuning, we average the last 10 checkpoints for every single model of the specific domain.

For evaluation, we adopt the cased BLEU scores calculated with SacreBLEU (Post, 2018).

4.2 Pre-processing and Post-processing

In pre-processing, we conduct data filtering, tokenization, subword encoding. For Chinese sentences, we use the DiDi tokenizer for tokenization. For English data, we do punctuation normalization and use Spacy[4] tokenizer for tokenization. We filter parallel sentences as described in Section 3.1. Finally, we collect a preprocessed bilingual training data consisting of 10M parallel sentences and 20M synthetic sentences. We adopt subword encoding for Chinese → English. Specifically, we learn a BPE with 40K merge operations, in which 37.8K and $27.8K$ subword tokens are adopted as Chinese and English vocabularies separately.

In the post-processing phase, we conduct unknown (UNK) words replacement, de-tokenization, punctuation, and numerals normalization. UNK words are simply removed in the sentences. We use the Moses scripts to true-case and de-tokenize the English translations.

4.3 Chinese → English

We adopt methods in Section 3 for Chinese → English task. Firstly we adopt techniques of iterative back-translation and knowledge distillation for generating synthetic parallel data based on monolingual data. We combine the synthetic data and bilingual data as the training data and randomly split training data into 6 portions and do experiments to obtain 3 most effective portions. We train several models with different initialization seeds, different training datasets, and different architectures with the sampled synthetic data and bilingual data. In this way, we obtain models with diversity. After that, we fine-tune the model with different in-domain data. Next, we do the model ensemble by exploiting the translation domain style and choose the best model on development data as the final submission. Here we use WMT 2018 test set and WMT 2019 test set as our development data. Finally, we adopt several re-ranking and post-processing methods to obtain the final submission.

Table 1 shows the results on WMT 2018 test data of Chinese → English. As shown in the table, data augmentation with iterative back-translation and knowledge distillation consistently improve the BLEU score. Fine-tuning with selected in-domain corpus plays an important role in our system, which helps achieve improvement about more than a 2 BLEU score. We observe that ensemble with log-

[4]https://github.com/explosion/spaCy

avg strategy achieves slight improvement, which may be caused by the conflicts between different topic domains. To alleviate domain conflicts, we incorporate the domain style information, which achieves 2.15 improvement over the best single model. We also observe a relatively slight improvement with re-ranking. The reason may be that we use the training data to train both the re-ranker and the NMT models, which produces similar scores while dealing with the same sentences. Similar conclusions can be drawn from Table 2.

Table 3 shows the BLEU comparisons with related works on the WMT 2018 and WMT 2019 test sets. From the table, we observe that our system achieves the best performance on the WMT 2018 test set and the second best performance on the WMT 2019 test set. This demonstrates the effectiveness of the proposed system.

In our final submission, the model is an ensemble of 6 models, including 2 Transformer, 1 Transformer with relative position attention, 2 Transformer with larger FFN size, and 1 Transformer with reversed source. We do translation with beam size=10 and length penalty=1.4. Finally, we achieve a cased BLEU score of 36.6 in WMT 2020 Chinese $\rightarrow$ English competition.

5 Conclusion

In this paper, we present our NMT systems for WMT2020 news translation shared tasks in Chinese $\rightarrow$ English translation direction. Our final system achieves substantial improvement over baseline systems by integrating the following techniques:

1. Data filtering
2. Data augmentation, including iterative back-translation, knowledge distillation, etc.
3. Fine-tuning with in-domain data
4. Model ensemble and leverage domain topic information

As a result, our submitted system achieves a 36.6 BLEU score in the Chinese $\rightarrow$ English direction of WMT 2020 news translation shared tasks.

References

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. 2011. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 355–362.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Colin Cherry and George Foster. 2012. Batch tuning strategies for statistical machine translation. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 427–436. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Kevin Duh, Graham Neubig, Katsuhito Sudoh, and Hajime Tsukada. 2013. Adaptation data selection using neural language models: Experiments in machine translation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 678–683.

Chris Dyer, Victor Chahuneau, and Noah A Smith. 2013. A simple, fast, and effective reparameterization of IBM model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500.

Markus Freitag, Yaser Al-Onaizan, and Baskaran Sankaran. 2017. Ensemble distillation for neural machine translation. *arXiv preprint arXiv:1702.01802*.

Xinze Guo, Chang Liu, Xiaolong Li, Yiran Wang, Guoliang Li, Feng Wang, Zhitao Xu, Liuyi Yang, Li Ma, and Changliang Li. 2019. Kingsoft's neural machine translation system for WMT19. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 196–202.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431. Association for Computational Linguistics.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *International Conference on Learning Representations*.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, System Demonstrations*.

Bei Li, Yinqiao Li, Chen Xu, Ye Lin, Jiqiang Liu, Hui Liu, Ziyang Wang, Yuhao Zhang, Nuo Xu, Zeyang Wang, et al. 2019. The Niutrans machine translation systems for WMT19. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 257–266.

Minh-Thang Luong and Christopher D Manning. 2015. Stanford neural machine translation systems for spoken language domains. In *Proceedings of the International Workshop on Spoken Language Translation*, pages 76–79.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 464–468.

Meng Sun, Bojian Jiang, Hao Xiong, Zhongjun He, Hua Wu, and Haifeng Wang. 2019. Baidu neural machine translation systems for WMT19. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 374–381.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Yingce Xia, Xu Tan, Fei Tian, Fei Gao, Di He, Weicong Chen, Yang Fan, Linyuan Gong, Yichong Leng, Renqian Luo, et al. 2019. Microsoft Research Asia's systems for WMT19. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 424–433.

Boliang Zhang, Ajay Nagesh, and Kevin Knight. 2020. Parallel corpus filtering via pre-trained language models. *arXiv preprint arXiv:2005.06166*.

Zhirui Zhang, Shujie Liu, Mu Li, Ming Zhou, and Enhong Chen. 2018. Joint training for neural machine translation models with monolingual data. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Facebook AI's WMT20 News Translation Task Submission

Peng-Jen Chen, Ann Lee, Changhan Wang, Naman Goyal
Angela Fan, Mary Williamson, Jiatao Gu
Facebook AI
{pipibjc,annl,changhan,namangoyal}@fb.com
{angelafan,marywilliamson,jgu}@fb.com

Abstract

This paper describes Facebook AI's submission to WMT20 shared news translation task. We focus on the low resource setting and participate in two language pairs, Tamil ↔ English and Inuktitut ↔ English, where there are limited out-of-domain bitext and monolingual data. We approach the low resource problem using two main strategies, leveraging all available data and adapting the system to the target news domain. We explore techniques that leverage bitext and monolingual data from all languages, such as self-supervised model pretraining, multilingual models, data augmentation, and reranking. To better adapt the translation system to the test domain, we explore dataset tagging and fine-tuning on in-domain data. We observe that different techniques provide varied improvements based on the available data of the language pair. Based on the finding, we integrate these techniques into one training pipeline. For En → Ta, we explore an unconstrained setup with additional Tamil bitext and monolingual data and show that further improvement can be obtained. On the test set, our best submitted systems achieve 21.5 and 13.7 BLEU for Ta → En and En → Ta respectively, and 27.9 and 13.0 for Iu → En and En → Iu respectively.

1 Introduction

We participate in the WMT20 news translation task in two low resource language pairs (four directions), Tamil ↔ English (Ta → En and En → Ta) and Inuktitut ↔ English (Iu → En and En → Iu). These language pairs are challenging due to the lack of in-domain bitext training data and limited monolingual data. For Tamil, the available bitext corpora are from various sources; however, none of the sources is in the news domain, and each corpus is in limited size or noisy. Inuktitut encompasses the challenges present for Tamil, but is even more challenging because the quantity of available monolingual data is even less than the bitext data.

We explore techniques that leverage available data from all languages. First, we investigate supervised learning together with pre-training using mBART (Liu et al., 2020). Second, inspired by the recent success of improving low resource languages through multilingual models (Arivazhagan et al., 2019; Tang et al., 2020), we explore the utility of multilingual models, in the form of multilingual pretraining and subsequent fine-tuning. Third, we leverage the monolingual data of the source and target languages using data augmentation techniques, such as back-translation (Sennrich et al., 2015) and self-training (Ueffing, 2006; Zhang and Zong, 2016; He et al., 2019). Following Chen et al. (2019), we apply these techniques iteratively. Fourth, we use noisy-channel model reranking (Yee et al., 2019) to further boost performance. The reranking uses language modeling to select a more fluent hypothesis, which requires monolingual data in the target language.

Additionally, we investigate how adding substantially more unconstrained data can further improve the performance of En → Ta system. We incorporate data from bitext mining efforts such as CCMATRIX (Schwenk et al., 2019) and CCALIGNED (El-Kishky et al., 2019), as well as additional monolingual data from CCNET (Wenzek et al., 2019) curated from CommonCrawl. The additional data is used for iterative back-translation and to train stronger language models for noisy-channel reranking.

In a complementary direction, we investigate ways to adapt the translation system to the target domain. We explore controlled generation by adding dataset tags to indicate domain. Furthermore, we fine-tune our system on the in-domain data.

For all language directions, we obtain our final systems by fusing a combination of the tech-

Proceedings of the 5th Conference on Machine Translation (WMT), pages 113–125

niques mentioned above. We observe that the bulk of the improvements in our systems are from iterative back-translation and self-training, except the En → Iu system where we only have exceptionally limited quantities of Inuktitut monolingual data. Noisy-channel reranking provides further improvement on top of strong systems, especially for to-English directions where we have high-quality news-domain monolingual data to train a good language model. Each of the other techniques, including dataset tagging, fine-tuning on in-domain data, and ensembling also provides nice improvements.

2 Data

For the constrained track, we use monolingual data from all languages provided in WMT20 for mBART pre-training (Liu et al., 2020), and we use bitext data between English and other languages for training the system from scratch or fine-tuning the pretrained mBART models. We also require English, Tamil, and Inuktitut monolingual data for techniques such as back-translation, self-training, and creating language models for noisy-channel reranking. For low resource languages, Tamil and Inuktitut, we use all the available monolingual data, e.g. NewsCrawl + CommonCrawl + Wikipedia dumps for Tamil, and CommonCrawl for Inuktitut. For English, we only use NewsCrawl as the monolingual data because it is sufficiently large, high-quality, and in the news domain.

For the unconstrained track, we use Tamil monolingual data and Tamil-English mined bitext data from external sources based on CommonCrawl. The details are described in Section 2.2.

2.1 Data filtering

2.1.1 Bitext data

For each data source for each language pair, we remove duplicate sentence pairs and use `fastText` (Joulin et al., 2016a,b) language identification to remove sentence pairs where either the source or the target sentence is not predicted as the expected language. The resulting size of the bitext data of each language pair is shown in Appendix Table A.1.

2.1.2 Monolingual Data

We use monolingual data after `fastText` language identification filtering from all languages provided in WMT20 to train our mBART model. CommonCrawl contains a large quantity of data, but is also quite noisy as it is crawled from the web. Furthermore, the sentences are not in the news domain. To clean the data and select the sentences closer to the news domain, we apply the in-domain filtering method described in (Moore and Lewis, 2010) for languages that have NewsCrawl monolingual data. First, we train two n-gram language models (Heafield, 2011) on NewsCrawl and CommonCrawl respectively. Then, for each sentence from CommonCrawl, we obtain scores from these two language models, compute the difference between normalized log-probability, and we remove the lowest-scoring sentences. We heuristically examine the data and remove the bottom 30%-60% of sentences. Concretely, the scoring function is $H_{NC}(s) - H_{CC}(s)$, where s is the sentence, $H_{NC}(s)$ and $H_{CC}(s)$ are the word-normalized cross entropy scores for sentence s by n-gram language model trained on NewsCrawl and CommonCrawl data respectively.

We concatenate sentences from different sources and remove duplicate sentences for each language. We show the detailed dataset statistics in Appendix Table A.2.

2.2 Unconstrained setup for Tamil

In the unconstrained track, additional data can be used. We incorporate two additional sources of data: noisy bitext from data mining and monolingual data.

2.2.1 Mined bitext data

We use mined bitext data from CCMATRIX (Schwenk et al., 2019) and CCALIGNED (El-Kishky et al., 2019), two complementary mining strategies. Both approaches use the web data from unconstrained CommonCrawl to identify noisy bilingual matched pairs. CCMATRIX embeds monolingual sentences using `LASER` (Schwenk and Douze, 2017) multilingual sentence embeddings. To identify matching bitext pairs, the distance from each sentence to each other sentence is calculated based on the distance in the embedding space. For CCALIGNED, documents that could correspond to bitext pairs are aligned first at the document level, then at the paragraph level, and finally at the sentence level. In total, we include 2M aligned English-Tamil mined sentences.

2.2.2 Monolingual data

We used additional Tamil monolingual data from CommonCrawl snapshots between 2017-26 to 2020-10 extracted by CCNET (Wenzek et al., 2019). We break down the document-level structure from CCNET into sentences and apply further processing. We concatenate all the snapshots of the additional monolingual data, deduplicate the sentences, apply `fastText` language identification and remove sentences are not predicted as Tamil. The final data results in 125M sentences. Subsequently, we concatenate the unconstrained monolingual data with constrained monolingual data, and we use them for back-translation and training Tamil language model.

3 System overview

We use the Transformer (Vaswani et al., 2017) as our model architecture for all of our systems. To better train models with datasets in different sizes, we use random search to select the hyper-parameters that achieve the best BLEU score on the validation set. We use sentencepiece (Kudo and Richardson, 2018) to learn the subword units to tokenize the sentences. The details of selected hyper-parameters are listed in Appendix D. All our systems are trained with `fairseq`[1] (Ott et al., 2019).

3.1 Dataset tag

Training and decoding the model with an indication of domain (such as a specified dataset tag) (Kobus et al., 2016) is a technique that allows us to control the output domain of the trained system. Similarly, Caswell et al. (2019); Chen et al. (2019) show that adding specific tag to back-translated and self-translated data can improve model performance. We add dataset tags to all of our systems described in this paper, by pre-pending a domain specific tag to the source sentence during training. At test time, we sweep over all the possible tags that are used during training including "no tag", and we choose the tag that achieves the best BLEU score on validation set. We find that when training with dataset tag, the supervised systems are 0.9 and 0.5 BLEU score higher than the system trained without dataset tag for Ta → En and En → Ta respectively. See results in Table 1.

[1] https://github.com/pytorch/fairseq

3.2 Baseline systems

We investigate a variety of baseline approaches as the starting point for our models. For both Tamil and Inuktitut languages, we explore four different baseline systems, (1) bilingual supervised, (2) multilingual supervised, mBART pretraining with (3) bilingual and (4) multilingual fine-tuning. These systems are trained with constrained bitext and monolingual data. We will then improve these baseline models, as described in subsequent sections.

3.2.1 Bilingual supervised

To train the base bilingual systems, we pre-pend the dataset tag to the source sentence to differentiate data from different corpus and concatenate all data sources for that language.

3.2.2 Multilingual supervised

Arivazhagan et al. (2019) shows that multilingual model can improve the model performance of medium and low resource languages, as multilingual models are often trained on greater quantities of data compared to bitext models. Thus, we investigate if multilingual supervised models can be stronger starting points. We use all the bitext data between English and other languages provided in WMT20 to train many-to-one (XX → English) and one-to-many (English → XX) models. One challenge of multilingual training is different language directions have different quantities of data, and the high resource language can starve for capacity while low resource language can benefit from the transfer. To balance the trade-off between learning and transfer, we follow Arivazhagan et al. (2019) with a temperature-based strategy to sample sentences from different languages. Furthermore, for each direction, we optimize the transfer by selecting the best temperature and model checkpoint based on the BLEU score of the target language pair validation set.

3.2.3 mBART-pretraining with bilingual and multilingual fine-tuning

For mid and low resource languages, the quantity of available bitext may be low, but large resources of monolingual data exist. This monolingual data can be used in the form of pre-training, followed by subsequent fine-tuning into translation models. We use mBART (Liu et al., 2020) – a multilingual denoising pre-training approach – to pre-train our systems, which has shown substantial improvements com-

pared to training the model from scratch. First, we pre-train mBART across 13 languages (Cs, De, En, Fr, Hi, Iu, Ja, Km, Pl, Ps, Ru, Ta, Zh) on all monolingual data provided by WMT 20. For pretraining, we used a batch size of 2048 sequences per batch and trained the model for $240K$ steps. We learn the SPM jointly on all languages. We sample the same amount of sentences from monolingual data of all languages to learn a vocabulary of $130,000$ subwords. In the fine-tuning stage, we use exactly the same data sources as the bilingual supervised model and multilingual supervised model. For multilingual fine-tuning, previously people have built bitext translation systems by fine-tuning pretrained mBART models. Recent work Tang et al. (2020) extended this to multilingual fine-tuning, which can create multilingual translation models from multilingual pre-trained models. Different from Tang et al. (2020), we tune the temperature rate separately for the four language directions we focus on. In the multilingual fine-tuning stage, we use random search to sweep over dropout, learning rate, and temperature sampling factor, and we select the model checkpoint based on the BLEU score evaluated on the target language pair validation set.

3.3 Iterative back-translation (BT)

Back-translation (Sennrich et al., 2015) is an effective data augmentation technique to improve model performance with target side monolingual data. The method starts from training a target to source translation system, which is subsequently used to translate the monolingual data in the target language back to source language. Then the synthetic back-translated dataset is concatenated with the raw bitext data to train the source to target translation model. After the source to target model is improved, the same technique can be applied again to train the back-translation system in the reversed direction. We repeat the process for several iterations until no significant improvement is obtained.

In all of our back-translation systems, we follow Chen et al. (2019) to add dataset tags to both raw bitext data and back-translated data. We upsample the bitext data, and the upsampling ratio is selected based on parameter sweeping and validating the resulting improvement on the validation set. Beam search with beam size 5 is used when generating the synthetic sentences.

3.4 Noisy-channel reranking (NCD)

Reranking is a technique that uses a separate model to score and better select hypotheses from the n-best list generated by the the source to target model. To rerank our system output, we use the noisy-channel model (Yee et al., 2019) as the scoring model (Ng et al., 2019; Chen et al., 2019). Given a source and target sentence pair (x, y), the noisy-channel model scores it with

$$\log P(y|x) + \lambda_1 \log P(x|y) + \lambda_2 \log P(y) \quad (1)$$

where $\log P(y|x)$, $\log P(x|y)$ and $\log P(y)$ are the forward model, backward model and language model scores. The weights, λ_1 and λ_2, are tuned through random search on the validation set. All of our submitted test set hypotheses are ranked and selected by noisy-channel reranking.

The language models used in noisy-channel reranking are Transformers. For constrained track, we use the monolingual data as described in Section 2 to train the language models for English, Tamil. For Inuktitut, we find that the monolingual data is very limited and even smaller than the size of bitext data, therefore we concatenate the CommonCrawl data with the Inuktitut side of the bitext data together to train the Inuktitut language model. For unconstrained Tamil language model, we train on the constrained data with the additional unconstrained data extracted by CCNET as described in Section 2.2. The SPM size, model hyper-parameters, and evaluation of the language models can be found in Appendix B.

3.5 Self-training (ST)

Self-training (Ueffing, 2006; Zhang and Zong, 2016; He et al., 2019) is a method that leverage monolingual data in source language to improve the system performance. We use the trained source to target translation system to translate monolingual data in source language to target language. Similar to BT, the synthetic dataset can be concatenated with bitext data to train the source to target model again. We follow Chen et al. (2019) and use the noisy-channel model to select the top synthetic sentence when decoding from monolingual data into the source language. We inject the same types of noise to the source side of synthetic data as He et al. (2019).

Shen et al. (2019); Chen et al. (2019) both show that self-training can provide complementary improvement in addition to back-translation, espe-

Model	Ta → En	En → Ta	Iu → En	En → Iu
w/o tag	15.6	8.5	31.4	16.1
with tag	16.5	9.0	31.3	16.1

Table 1: Systems trained with and w/o dataset tags. The BLEU score is reported on validation set. We sweep all available dataset tags when decoding on validation set and report the best performing dataset tag. The BLEU scores of each dataset tag are reported in Appendix C

cially when (1) there is lack of target side monolingual data, (2) source side monolingual data is much similar to the domain of test set compared with target side monolingual data, and (3) the decoding method outperforms greedy decoding on the source to target model. Therefore, we experiment self-training on En → Iu due to greater quantities of in-domain source side monolingual data, on Iu → En in Nunavut Hansard domain with Inuktitut side of bitext data due to much more in-domain monolingual data on the source side, and on Ta → En because we observe great improvement from noisy-channel reranking. However, we only observe significant improvement on Ta → En system.

3.6 Fine-tuning (FT) on validation set

Fine-tuning is a technique to adapt the model to the target domain when the initial model is not trained with training data in the target domain. In both Tamil and Inuktitut, none of the training data is in news domain as the test data, therefore we fine-tune our final systems on a portion of the validation data and evaluate on the rest of hold-out validation data. For Tamil systems, we split the validation data with a 75-25 split, where 75% of the data is used for fine-tuning and 25% of the data is used for evaluation. Ta → En and En → Ta systems are fine-tuned and evaluated on the same split of validation dataset. For Inuktitut systems, we split the validation set based on the domain — Nunavut Hansard or news. For each domain, we split the validation data with a 75-25 split for fine-tuning and evaluation. We fine-tune our best performing Iu → En and En → Iu systems in domain on the corresponding validation set split.

4 Results

In this section, we describe the details of our systems, and we report SACREBLEU (Post, 2018) on the validation set for intermediate iterations and ablations. For our validation set fine-tuned systems, we report the BLEU score on our validation holdout set split. Our general strategy for all language directions was to identify the best performing baseline setting, then iteratively improve upon the baseline using back-translation and self-training. Finally, we apply noisy-channel reranking and fine-tuning on validation set to create our final submission.

4.1 Baseline

We explore four different baseline approaches as described in Section 3.2 for each language direction in the constrained setup, Inuktitut ↔ English and Tamil ↔ English. The detailed results are shown in Table 2.

First, bilingual models are trained with bilingual bitext data. Next, we focus on multilingual training. The multilingual supervised models are trained with all the available bitext data provided by WMT20. We use the same SPM as described in Section 3.2.3. For both bilingual and multilingual models, we initialize the model weights either randomly or with pre-trained mBART model weights. Therefore, for each language direction, we have four combinations, bilingual supervised, multilingual supervised, mBART + bilingual fine-tuning and mBART + multilingual fine-tuning. We use dataset tags for all systems, and we sweep the tag that performs the best when decoding on the validation set. Additional details and hyper-parameters are provided in the Appendix D.

For to-English directions, both multilingual models and mBART pretraining can get better model performance than bilingual supervised model as shown in Table 2. For Ta → En direction, mBART + multilingual fine-tuning performs the best with 20.4 BLEU, which outperforms bilingual supervised system by 3.2 BLEU score. For the Iu → En direction, mBART + bilingual fine-tuning works the best and gets 32.9 BLEU score, which outperforms bilingual supervised baseline by 2.8 BLEU score. However, for from-English directions, we do not observe similar advantages with either multilingual model or mBART pretraining, and a properly tuned bilingual supervised model achieves the best results for both directions. We get 8.0 BLEU score for En → Ta direction, and we get 16.1 BLEU score for En → Iu direction.

4.2 Tamil systems

4.2.1 Constrained Ta → En system

For the Ta → En system, we first use the En → Ta bilingual baseline system (ensemble) to gener-

System	Ta → En	En → Ta	Iu → En	En → Iu
bi.	17.2	8.0	29.7	16.1
multi.	18.2	7.1	30.7	15.8
bi-FT*	18.9	8.0	32.9	16.1
multi-FT*	20.4	7.4	32.5	16.0

Table 2: BLEU scores of baseline systems evaluated on the validation set. * Pre-trained on mBART.

ate back-translation data from English NewsCrawl data. We then train our first iteration back-translation system ("iter1-BT") with upsampled bitext (upsampling ratio tuned on the validation set). Similarly, we train our second iteration back-translation system ("iter2-BT") with upsampled bitext and back-translation data generated by En → Ta iter1-BT system (ensemble). The iter2-BT system (ensemble) is then used to generate ST data from Tamil NewsCrawl, CommonCrawl and Wiki data. We combine it with iter2-BT system's data to train the iter2-BT+ST system. Finally, we fine-tune this system on the validation set and apply noisy-channel reranking to select the hypotheses. We explore Transformer models of different capacities and choose Transformer *big* (with 8K feed-forward dimension) for a good balance of performance and training speed. For the iter2-BT+ST system (and its ensemble/finetuned version), we further enlarge the encoder to 10 layers given higher data abundance. We can see from Table 3 that our training pipeline improves model performance steadily (≥ 1.3 validation BLEU) after iterations, and in-domain fine-tuning as well as noisy-channel reranking are very helpful to alleviate the effects of train-test domain mismatch.

4.2.2 Constrained En → Ta system

For the En → Ta system, we first use the mBART+multi-FT baseline system for Ta → En to generate back-translation data from the monolingual data. We add different back-translation dataset tags based on the source of monolingual data and train our first iteration back-translation system ("iter1-BT") by tuning upsampling ratios on the bitext and back-translation datasets. For the model architecture, we explore the options of training Transformers from scratch and fine-tuning a pretrained mBART model and find that the former performs better with ensembles. Doing one iteration of training with back-translation data gives 5.8 BLEU increase (Table 3). We further train the second iteration back-translation system ("iter2-BT") with back-translation data generated from the best iter1-BT Ta → En system. As the gain from the second iteration is small (0.4 BLEU), we do not continue for the third iteration. Noisy-channel reranking is applied with the best systems from both language directions and the Tamil language model (Appendix B). We observe little gain (0.1 BLEU) and suspect it's due to the high perplexity of the language model. Further fine-tuning the iter2-BT model on the validation set gives 4.1 BLEU score improvement on the validation holdout set.

System	Ta → En	En → Ta
baseline	20.4	8.0
+ ensemble	21.2	9.0
iter1-BT	23.4	13.8
+ ensemble	24.8	14.1
iter2-BT	25.6	14.2
+ ensemble	26.4	14.3
+ NCD	28.5	14.4
eval on valid holdout		
iter2-BT	26.2	14.6
iter2-BT+ST	27.5	-
iter2+FT on valid	28.0	18.7
+ ensemble	28.3	19.0
+ NCD	29.8	19.5
unconst. eval on valid holdout		
iter2-BT	-	15.2
iter2-BT+FT	-	19.6
+ ensemble	-	19.6
+ NCD	-	20.2

Table 3: Results of Tamil systems. We report the BLEU scores on newsdev2020 validation set.

system	Iu → En		
	NH	News	Combined
baseline	42.4	19.2	32.9
+ ensemble	42.4	19.4	32.9
iter1-BT	43.3	24.1	35.1
+ ensemble	43.8	24.6	35.7
eval on valid holdout			
iter1-BT	46.1	24.3	35.0
iter1-BT+FT on valid	47.3	31.1	38.4
+ ensemble	48.2	31.7	39.2
+ NCD	49.0	32.8	40.2

Table 4: Results of Iu → En systems. We report BLEU scores on both domain-split and the whole newsdev2020 validation set

4.2.3 Unconstrained En → Ta system

For the unconstrained track, we first used the iteration1 + back-translation ensemble model to back-translate the additional monolingual data from CommonCrawl. Subsequently, we combined

system	En → Iu		
	NH	News	Combined
baseline	24.5	5.3	16.1
+ ensemble	24.8	5.6	16.3
iter1	24.8 (ST)	5.5 (BT)	16.3
+ ensemble	25.0 (ST)	5.8 (BT)	16.5
eval on valid holdout			
iter1	27.6 (ST)	5.4 (BT)	15.5
iter1+FT on valid	28.9	14.5	20.8
+ ensemble	28.9	15.1	21.1
+ NCD	28.9	16.6	22.0

Table 5: Results of En → Iu systems. We report BLEU scores on both the domain-split and whole newsdev2020 validation set.

back-translated data from unconstrained monolingual sources with back-translated data from WMT monolingual data from English and Tamil, with the WMT bitext and mined Ta → En data. We used the same BPE and vocabulary as the constrained system. The data was deduplicated, and the validation and test data removed if an exact match was present in the training data. The mined data was additionally cleaned to remove sentences longer than 250 BPE tokens, as well as bitext pairs where the length between the source and target was greater than 2.5x difference. Subsequently, we trained a large Transformer sequence-to-sequence model on the total combined data using various data domain tags. After training was complete, we further fine-tuned on the validation set, as described in Section 4.2.2. We applied noisy-channel reranking when decoding test data. The forward model is ensembled with two of the best performing fine-tuned models. The backward model is the best performing model in Section 4.2.1, which is ensembled with two fine-tuned models. The language model is unconstrained Tamil language model described in Section 3.4. We rerank from best 20 hypothesises generated by ensembled forward model, and we achieve 20.2 BLEU score on validation set.

4.3 Inuktitut systems

The Inuktitut validation and test set are composed of data from two different domains, the proceeding of the Legislative Assembly of Nunavut from Nunavut Hansard (NH) and news. We find that the model can be further improved if we optimize our translation training pipeline for these two domains separately, and therefore we train and report BLEU score separately for each domain. We also report the BLEU score on the whole validation set, where we use the domain-specific system to decode on the portion of the corresponding domain, concatenate the hypothesises and compute the BLEU score.

4.3.1 Constrained Iu → En systems

For the Iu → En system, we use En → Iu bilingual supervised system described in Section 4.1 for back-translation. The model used for decoding is an ensemble of 3 En → Iu models, and we decode from the English NewsCrawl data. We concatenate the back-translated data with bitext data and sweep the upsampling ratio of the bitext data to find the best ratio. We experiment with both mBART pretraining + bilingual fine-tuning and training from scratch, and we find that mBART + bilingual fine-tuning works better on Nunavut Hansard domain of validation set, and training from scratch works better on news domain. The hypothesis is that the English NewsCrawl monolingual data for back-translation is in-domain with the news domain validation set and there is huge amount of English NewsCrawl data, so the advantage of pretraining is not significant. We also experiment with self-training on Iu → En direction in Nunavut Hansard domain, where we use the source to target model (ensembled) to decode from the Inuktutit side of Nunavut Hansard 3.0 parallel corpus with noisy-channel reranking; however, we do not observe any improvement. The best result at the first iteration is from the back-translation system, which outperforms baseline system by 2.2 BLEU score (Table 4), where most of the gain comes from improvement on news domain.

We do not observe gains for doing the second iteration of back-translation for Iu → En system, and we suspect that it is due to lack of improvement for our En → Iu model from supervised approach to the first iteration. We then fine-tune the best iteration 1 Iu → En models on validation data for each domain. The final domain-specific systems are ensembled from the fine-tuned models and followed by noisy-channel reranking. To use noisy-channel reranking for Nunavut Hansard domain, we fine-tune the English language model described in 3.4 on English side of the Nunavut Hansard 3.0 training data provided in WMT20. The best Iu → En system we submit has 40.2 BLEU score on our validation holdout set.

4.3.2 Constrained En → Iu systems

We experiment with both self-training and back-translation with the best baseline systems reported in 4.1 to improve En → Iu system. For self-training,

we use ensembled supervised En → Iu model and beam decoding with beam size 5 to decode from English monolingual data. We decode from the English side of Nunavut Hansard 3.0 parallel corpus to train the model for Nunavut Hansard domain, and we decode from the English NewsCrawl data for news domain. However, we do not observe improvement for news domain, and there is only mild improvement (0.3 BLEU) for Nunavut Hansard domain as shown in Table 5. For back-translation, we use iteration 1 Iu → En news domain model from 4.3.1 to decode constrained Inuktitut CommonCrawl data. We get no improvement on Nunavut Hansard domain and mild improvement (0.2 BLEU) on news domain. We use self-training system for Nunavut Hansard domain and back-translation system for news domain, and it achieves 16.3 BLEU score on the validation set, which is merely 0.2 BLEU score improvement over baseline system. We then fine-tune the best systems we get on domain-specific validation set splits, followed by ensembling and noisy-channel reranking. The fine-tuning is very effective for the news domain, where we get 9.1 BLEU score improvement. This is expected because we do not have any training data from news domain. Our final submitted system achieves 22.0 on our validation holdout set.

Submitted system	BLEU
Ta → En	21.5
En → Ta	12.6
En → Ta (unconst.)	13.7
Iu → En	27.9
En → Iu	13.0

Table 6: Results of our best submitted systems of each direction. We report BLEU scores on newstest2020.

5 Conclusion

This paper describes Facebook AI's Transformer based translation systems for the WMT20 news translation shared task. We focused on two low-resource languages pairs, Tamil ↔ English and Inuktitut ↔ English, and we explored the same set of techniques, including dataset tagging, mBART pretraining and fine-tuning, back-translation and self-training, fine-tuning on domain-specific data, ensembling, and noisy-channel reranking. We demonstrated strong improvements by stacking these techniques properly on three language directions, Ta → En , En → Ta , and Iu → En . The En → Iu direction is difficult to improve due to lack of target side monolingual data. Surprisingly, self-training does not work on En → Iu either even we have huge amounts of in-domain English side monolingual data. We are interested in continued exploration on how to better leverage source side monolingual data to improve En → Iu and other low resource languages where we do not have enough target side monolingual data.

6 Acknowledgements

We thank Marc'Aurelio Ranzato for providing discussion and guidance during the competition, Vishrav Chaudhary for sharing insightful data cleaning approaches, Guillaume Wenzek for previous work on ccNET for monolingual data used in unconstrained setting, Yuqing Tang for the work of mBART pretraining and multilingual fine tuning, Ahmed El-Kishky and Holger Schwenk for sharing their mined data for Tamil, Sergey Edunov for sharing cleaned up dataset to speed up our early exploration, and Michael Auli for sharing experience about noisy-channel reranking technique.

References

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George Foster, Colin Cherry, et al. 2019. Massively multilingual neural machine translation in the wild: Findings and challenges. *arXiv preprint arXiv:1907.05019*.

Isaac Caswell, Ciprian Chelba, and David Grangier. 2019. Tagged back-translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 53–63, Florence, Italy. Association for Computational Linguistics.

Peng-Jen Chen, Jiajun Shen, Matt Le, Vishrav Chaudhary, Ahmed El-Kishky, Guillaume Wenzek, Myle Ott, and Marc'Aurelio Ranzato. 2019. Facebook AI's WAT19 Myanmar-English translation task submission. *arXiv preprint arXiv:1910.06848*.

Ahmed El-Kishky, Vishrav Chaudhary, Francisco Guzman, and Philipp Koehn. 2019. A massive collection of cross-lingual web-document pairs.

Junxian He, Jiatao Gu, Jiajun Shen, and Marc'Aurelio Ranzato. 2019. Revisiting self-training for neural sequence generation. *arXiv preprint arXiv:1909.13788*.

Kenneth Heafield. 2011. KenLM: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland. Association for Computational Linguistics.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016a. Fasttext.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016b. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations (ICLR)*.

Catherine Kobus, Josep Maria Crego, and Jean Senellart. 2016. Domain control for neural machine translation. *CoRR*, abs/1612.06140.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226*.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation.

Robert C. Moore and William Lewis. 2010. Intelligent selection of language model training data. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 220–224, Uppsala, Sweden. Association for Computational Linguistics.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook FAIR's WMT19 news translation task submission. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 314–319, Florence, Italy. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

Holger Schwenk and Matthijs Douze. 2017. Learning joint multilingual sentence representations with neural machine translation. *arXiv preprint arXiv:1704.04154*.

Holger Schwenk, Guillaume Wenzek, Sergey Edunov, Edouard Grave, and Armand Joulin. 2019. Ccmatrix: Mining billions of high-quality parallel sentences on the web. *arXiv preprint arXiv:1911.04944*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709*.

Jiajun Shen, Peng-Jen Chen, Matt Le, Junxian He, Jiatao Gu, Myle Ott, Michael Auli, and Marc'Aurelio Ranzato. 2019. The source-target domain mismatch problem in machine translation.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *IEEE conference on computer vision and pattern recognition*.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2020. Multilingual translation with extensible multilingual pretraining and finetuning.

Nicola Ueffing. 2006. Using monolingual source-language data to improve mt performance. In *International Workshop on Spoken Language Translation (IWSLT) 2006*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzman, Armand Joulin, and Edouard Grave. 2019. Ccnet: Extracting high quality monolingual datasets from web crawl data. *arXiv preprint arXiv:1911.00359*.

Kyra Yee, Yann Dauphin, and Michael Auli. 2019. Simple and effective noisy channel modeling for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5696–5701, Hong Kong, China. Association for Computational Linguistics.

Jiajun Zhang and Chengqing Zong. 2016. Exploiting source-side monolingual data in neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545.

A Constrained data

In this section, we list the statistics for all the constrained datasets we use to build for our systems.

Bitext data Table A.1 shows the bitext data we used for multilingual systems. We use all bitext data between English and other 11 languages provided in WMT 20 except a couple of sources. We do not include the data back-translated by other system to avoid introducing bias. We do not include CzEng 2.0 for Czech nor CCMT for Chinese due to human mistake. We follow the filtering steps described in Section 2.1.1, and the size of dataset for each language pairs are listed in Table A.1.

Monolingual data Table A.2 shows the list of monolingal data we use for mBART-pretraining with 13 languages. We follow Section 2.1.2 to filter the monolingual data, and we list the amount of data before and after the filtering step.

B Language model used in noisy-channel reranking

Language model is required in the noisy-channel reranking system. We learn the BPE subwords with sentencepiece, and we train the Transformer based causal language models with `fairseq` in fp16 mode. The model size and hyper-parameters are tuned based on the perplexity of newsdev2020 validation sets per language. We describe the data and hyper-parameters of each language below, and we report the perplexities in Table B.1.

English language model We train our English language model with the high quality NewsCrawl data provided by WMT 20. We use the same filtering steps in Section 2.1.2 for NewsCrawl. We learn the BPE with 32K vocabulary size. We train the transformer-based model with 36 transformer layers, 1280 embedding dimension size, 5120 ffn dimension size, 20 attention heads and resulting in 749M parameters. The optimizer is Adam (Kingma and Ba, 2015) optimizer with $beta1 = 0.9$ and $beta2 = 0.98$. We use polynomial decay learning rate scheduler with 0.005 learning rate and 0.1 dropout rate. The maximum tokens are 4096 for each batch per GPU, and we train with 64 GPUs for 58K updates. As we show in Table B.1, this model achieves 23.3 perplexity on English side of Ta-En newsdev2020 set, 25.3 perplexity on news portion of Iu-En newsdev2020 set, and 29.7 perplexity on Nunavut Hansard portion of Iu-En newsdev2020 set. The perplexity on news validation sets are lower than none-news validation set. We use the English language model to rerank Ta $\rightarrow$ En system and news domain of Iu $\rightarrow$ En system.

To better rerank Iu $\rightarrow$ En hypothesises for Nunavut Hansard domain, we fine-tune the English language model on English side of Nunavut Hansard 3.0 parallel corpus. The perplexity on Nunavut Hansard portion of Iu-En newsdev2020 set is significantly improved from 29.7 to 8.1. We use the fine-tuned English language model to rerank the Nunavut Hansard domain of Iu $\rightarrow$ En system.

Tamil language model We train the Tamil language model for constrained En $\rightarrow$ Ta system with all the available Tamil monolingual data preprocessed in Section 2.1.2. The BPE vocabulary size is 32K. We train the transformer-based language model with 24 transformer layers, 1024 embedding size, 4096 ffn embedding size, 16 attention heads and resulting in 335M parameters. We use Adam optimizer with $beta1 = 0.9$ and $beta2 = 0.98$. We use polynomial decay learning rate scheduler with 0.005 learning rate and 0.1 dropout rate. The maximum tokens are 8192 for each batch per GPU, and we train with 16 GPUs for 46K updates. The model achieves 61.8 perplexity on Tamil side of Ta-En newsdev2020 set.

For unconstrained En $\rightarrow$ Ta system, we use both constrained Tamil monolingual data and the additional Tamil monolingual data described in Section 2.2. We share the same 32K BPE vocabulary as constrained Tamil language model. We use a larger transformer model with 32 transformer layers, 1024 embedding size, 4096 ffn embedding size, 8 attention heads. We use Adam optimizer with $beta1 = 0.9$ and $beta2 = 0.98$. We use cosine learning rate scheduler with 0.0001 learning rate and 0.3 dropout rate. The maximum tokens are 3072 for each batch per GPU, and we train with 32 GPUs for 69K updates. The model achieves 40.6 perplexity on Tamil side of Ta-En newsdev2020 set, which is better than the constrained Tamil language model.

Inuktitut langauge model The Inuktitut language model is trained with Inuktitut side of Nunavut Hansard 3.0 parallel corpus and the constrained Inuktitut monolingual data provided by WMT 20. The BPE vocabulary size is 5K. We train the transformer-based language model with

6 transformer layers, 512 embedding size, 4096 ffn embedding size, 8 attention heads and resulting in 34M parameters. We use Adam optimizer with $beta1 = 0.9$ and $beta2 = 0.98$. We use inverse square root learning rate scheduler with 0.0005 learning rate and 0.3 dropout rate. The maximum tokens 2048 for each batch per GPU, and we train with 8 GPUs for 89K updates. The model achieves 34.9 perplexity on Nunavut Hansard domain of Iu-En newsdev2020 set, and 81.69 perplexity on news portion of Iu-En newsdev2020 set.

C The effect of dataset tag at decoding time

We train our systems with dataset tag, and we sweep the dataset tags by add different tags to the same validation set and select the best performing tag. Table C.1 and C.2 show the system performance across different dataset tags.

First, we observe that sweeping the best performing dataset tag at decoding time is necessary. Using "no tag" to decode works the best for both Ta → En and En → Ta systems; however, using specific dataset tags works better for Iu → En and En → Iu systems. Second, the large BLEU score variations when decoding with different dataset tags show that the tags help the model to better adapt to different domains.

Overall, systems trained with dataset tags works better than trained without dataset tag as we show in Table 1.

D Hyper-Parameters

In this section, we report the hyper-parameters we use. For all of our translation systems, we use transformer based encoder-decoder model with shared embedding across encoder, decoder input and output embedding. We use Adam optimizer with $beta1 = 0.9$ and $beta2 = 0.98$, inversed square root learning rate scheduler, and 4000 warm-up steps with linearly increased rate. The loss is cross-entropy with label smoothing (Szegedy et al., 2016). We use the same batch sizes with maximum number of tokens 4096, and all models are trained with fp16. We sweep other hyper-parameters with random search, and we select the best performing system based on the evaluated BLEU scores on validation sets.

mBART pretraining We train the denoising mBART model with the constrained monolingual data from 13 languages described Section 2.1.2. We learn joint BPE across all languages with vocabulary size 130K. The transformer based encoder-decoder model has 12 encoder and decoder layers, 1024 embedding dimension, 4096 ffn embedding dimension and 16 attention heads, resulting in 487M parameters. We train the model with 0.0003 learning rate, 0.1 dropout rate, and no label-smoothing. We train the model with 256 GPUs for 240K updates.

Tamil systems For Ta → En, the best performing systems are mBART+multilingual fine-tuning model for baseline system, back-translation system for iteration 1 and BT+ST system for iteration 2. We report the hyper-parameters of the best performing system at each iteration in Table D.1.

For En → Ta, the best performing systems are bilingual supervised model for baseline system, back-translation system for iteration 1 and iteration 2. We report the hyper-parameters of the best performing system at each iteration, including the unconstrained system in Table D.2.

Inuktitut systems For Iu → En, the best baseline system is the mBART pretraining with bilingual fine-tuning. In iteration 1, we tune the model separately for Nunavut Hansard domain and news domain. The best Nunavut Hansard domain model is mBART pretraining with bilingual fine-tuning on bitext and news back-translated data, and the best news domain model is the back-translation model train from scratch. For En → Iu, the best baseline system is bilingual supervised model. Similar to Iu → En system, we tune the model separately for Nunavut Hansard domain and news domain in iteration 1. The best system for Nunavut Hansard domain is self-training model train from scratch, and the best system for news domain is the back-translation model train from scratch. We report the hyper-parameters of the best performing Iu → En and En → Iu systems at each iteration in Table D.3 and D.4.

Language pair	# of sentences (M)		Missing datasets
	Raw	Cleaned	
Cs-En	9.3	8.6	CzEng2.0, back-translated news
De-En	48	45.9	
Hi-En	1.48	1.27	
Iu-En	0.77	0.77	
Ja-En	18.2	16.2	
Km-En	4.4	2.46	
Pl-En	11.6	10.6	
Ps-En	1.13	0.58	
Ru-En	43.5	32.8	back-translated news
Ta-En	0.71	0.62	
Zh-En	19.6	15.8	CCMT, back-translated news

Table A.1: En-XX bitext data used for bilingual and multilingual systems. For each language pair, we use all available sources released in WMT20 except the datasets that are listed in the table.

Language	# of sentences (M)		Sources
	Raw	Cleaned	
Cs	355	287	NCL, NC, CC
De	3528	1355	NCL, NC, EP, CC
En	4264	2685	NCL, NC, ND, EP, CC, Wiki
Fr	5853	1455	NCL, NC, ND, EP, CC
Hi	45	43.4	IITB
Iu	0.9	0.9	Nunavut Hansard parallel corpus 3.0, CC
Ja	1776	1182	NCL, NC, CC
Km	12.1	11.3	CC, Wiki
Pl	1459	1183	NCL, EP, CC
Ps	5.9	5.4	CC, Wiki
Ru	1261	665	NCL, NC, CC
Ta	30.3	29.4	NCL, CC, Wiki
Zh	1677	806	NCL, NC, CC

Table A.2: Monolingual data used for mBART pretraining and back-translation. The abbreviation in the sources column represent the following, CC: CommonCrawl, EP: Europarl, NC: NewsCommentary, NCL: NewsCrawl, ND: NewsDiscussions, Wiki: Wikipedia

Target language	Training data		BPE size	PPL on newsdev2020		
	source	# of sentences		Ta-En	Iu-En (NH)	Iu-En (news)
English	NewsCrawl	190M	32K	23.3	29.7	25.3
+ FT on English side of NH				77.6	8.1	27.1
Tamil	CommonCrawl, NewsCrawl, Wikipedia	30M	32K	61.8	-	-
unconst. Tamil	constrained Tamil data, CommonCrawl in Sec. 2.2	155M		40.6	-	-
Inuktitut	Inuktitut side of Nunavut Hansard 3.0, CommonCrawl	860K	5K	-	34.9	81.7

Table B.1: Statistics of language models for each language.

Tag	Ta → En	En → Ta
None	16.5	9.0
mkp	15.4	8.0
nlpc	15.6	6.8
pib	15.5	8.6
pmindia	15.5	8.7
tanzil	11.9	0.6
ufal	16.1	8.2
wikimatrix	4.0	6.4
wikititles	15.8	8.5

Table C.1: Tamil bilingual supervised single model performance when decoding on validation set with different dataset tags. The BLEU score is evaluated newsdev2020 validation set.

Tag	Iu → En	En → Iu
None	29.7	15.8
Nunavut Hansard	31.3	16.0
wikititles	30.1	16.1

Table C.2: Inuktitut bilingual supervised single model performance when decoding on validation set with different dataset tags. The BLEU score is evaluated on newsdev2020 validation set.

System	Subword (size)	# params	layers	embed size	ffn embed size	attention heads	learning rate	dropout	label smoothing	# GPUs
Baseline system (mBART+multi-FT)	BPE (130K)	487M	12	1024	4096	16	0.0001	0.2	0.2	16
iter1 (BT)	Unigram (16384)	293M	6	1024	8192	16	0.0005	0.1	0.1	8
iter2 (BT+ST)	Unigram (16384)	378M	10	1024	8192	16	0.001	0.2	0.2	64

Table D.1: Hyper-parameters of the best performing Ta → En systems.

System	Subword (size)	# params	layers	embed size	ffn embed size	attention heads	learning rate	dropout	label smoothing	# GPUs
Constrained Tamil										
Baseline system (bilingual supervised)	Unigram (16384)	31M	3	512	2048	8	0.0005	0.3	0.1	8
iter1 (BT)	BPE (20K)	314M	10	1024	4096	16	0.0007	0.3	0.3	8
iter2 (BT)	BPE (20K)	314M	10	1024	4096	16	0.0007	0.2	0.3	8
Unconstrained Tamil										
iter2 (BT)	BPE (20K)	1.2B	10	2048	8192	16	0.0001	0.3	0.1	8

Table D.2: Hyper-parameters of the best performing En → Ta systems.

System	Subword (size)	# params	layers	embed size	ffn embed size	attention heads	learning rate	dropout	label smoothing	# GPUs
Baseline system (mBART+bi-FT)	BPE (130K)	487M	12	1024	4096	16	3e-5	0.1	0.1	16
NH-domain: iter1-BT (mBART+bi-FT)	BPE (130K)	487M	12	1024	4096	16	1e-4	0.2	0.2	16
news-domain: iter1-BT	BPE (5K)	559M	12	1024	8192	16	0.001	0.2	0.2	64

Table D.3: Hyper-parameters of the best performing Iu → En systems.

System	Subword (size)	# params	layers	embed size	ffn embed size	attention heads	learning rate	dropout	label smoothing	# GPUs
Baseline system (bilingual supervised)	BPE (5K)	122M	4	1024	4096	8	0.001	0.3	0.3	4
NH-domain: iter1-ST	BPE (5K)	152M	5	1024	4096	16	0.0005	0.2	0.2	4
news-domain: iter1-BT	BPE (5K)	152M	5	1024	4096	16	0.001	0.2	0.2	4

Table D.4: Hyper-parameters of the best performing En → Iu systems.

Linguistically Motivated Subwords for English-Tamil Translation: University of Groningen's Submission to WMT-2020

Prajit Dhar **Arianna Bisazza** **Gertjan van Noord**

University of Groningen

{p.dhar, a.bisazza, g.j.m.van.noord}@rug.nl

Abstract

This paper describes our submission for the English-Tamil news translation task of WMT-2020. The various techniques and Neural Machine Translation (NMT) models used by our team are presented and discussed, including back-translation, fine-tuning and word dropout. Additionally, our experiments show that using a linguistically motivated subword segmentation technique (Ataman et al., 2017) does not consistently outperform the more widely used, non-linguistically motivated SentencePiece algorithm (Kudo and Richardson, 2018), despite the agglutinative nature of Tamil morphology.

1 Introduction

In this paper we present the neural machine translation (NMT) systems submitted to the WMT-2020 English-Tamil (EN→TA) news translation task. This task is challenging mainly for two reasons:

1. Differing syntax: English is an Indo-European language which is fusional and SVO (Subject-Verb-Object). On the other hard, Tamil is part of the Dravidian language family and is a SOV language that is agglutinative. A good NMT system is expected to discern the various morphological forms on the Tamil target side.

2. Scarcity of training data: Prior to WMT-2020, there existed only a few corpora for parallel EN-TA sentences (Ramasamy et al., 2012; Germann, 2001). This left us with the choice of either only utilizing the low amount of parallel sentences or finding out ways of artificially enlarging the training data.

Through our submission we wish to provide solutions to the following questions:

- Is linguistically motivated subword segmentation beneficial for EN-TA translation?
- Can the addition of TA monolingual data compensate for the small amount of parallel EN-TA sentences despite the domain mismatch?
- Can fine-tuning on a corpus of Indian news improve quality on the WMT news translation task?

We start our paper with a short description of the Tamil language before delving into the various techniques adopted by our submitted NMT systems.

2 Tamil Language

Tamil is a Dravidian language spoken by around 80 million people. Tamil morphology is agglutinative and suffixal, i.e words are formed by suffixing morphemes to a lemma (Annamalai et. al 2014, cited in Sarveswaran et al. (2019)). Tamil suffixes can be either derivational (marking a change in PoS and/or meaning) or inflectional. In particular, nouns in Tamil are inflected for number, gender, case and animacy while verbs are inflected for tense, mood, aspect, negation, interrogation, information about emphasis, speaker perspective, sentience or rationality, and conditional and causal relations. Table 4 shows examples of the case forms in singular for the noun புத்தகம ’book’.

All the aforementioned statements substantiate the fact that Tamil morphology is highly complex. In fact, Ramasamy et al. (2012) identified 716 inflectional rules for nouns and 519 rules for verbs. Furthermore, designing a translation system for Tamil is challenging given the lack of training data (compare the sizes of Japanese and Tamil parallel datasets in WMT 2020, both agglutinative, however having vastly different training data; 25M sentences and 630k, respectively).

Proceedings of the 5th Conference on Machine Translation (WMT), pages 126–133
Online, November 19–20, 2020.

3 Previous Work

One of the earliest automatic translation systems for English→Tamil was by Germann (2001). They created a hybrid statistical/rule-based machine translation (SMT) system and trained it on only 5k EN-TA parallel sentences. Ramasamy et al. (2012) created SMT systems (phrase-based and hierarchical) that were trained on a much larger dataset of 190k parallel sentences. They also performed pre-processing steps involving morphological rules based on Tamil suffixes that improved upon the BLEU score of the baseline model (from 9.42 to 9.77). Their dataset (henceforth called UFAL) became the default benchmark for EN-TA translation systems until 2019, and we also use it in our experiments as an additional (general-domain) development set.

To the best of our knowledge, there have only been a handful of NMT systems trained on EN→TA. For the Indic languages multilingual tasks of WAT-2018, Sen et al. (2018), Dabre et al. (2018) and Ojha et al. (2018) reported BLEU scores for EN→TA. The Phrasal-based SMT system of Ojha et al. (2018) with a score of 30.53 BLEU outperformed the NMT systems of Sen et al. (2018) (11.88) and Dabre et al. (2018) (18.60), suggesting that the NMT systems were not suitable for translating a highly morphological language such as Tamil. However, the following year, Philip et al. (2019) outperformed Ramasamy et al. (2012) on the UFAL dataset with a BLEU score of 13.05. They report that techniques such as domain adaptation and back-translation can make training NMT systems on low-resource languages possible.

4 Datasets

For our constrained systems, we restrict ourselves to the datasets provided by WMT.

Parallel Table 1 presents the various parallel corpora along with their size and genre. The various corpora come from various sources and differ considerably in size. We also observe a very large difference in number of tokens between the two languages, with around 5 times more English tokens than Tamil tokens.

Monolingual Table 2 presents the monolingual Tamil corpora used in our experiments. Monolingual data is about 3 times larger than the parallel data in terms of tokens.

4.1 Pre-processing

For both parallel and monolingual data, the following steps are carried out sequentially:

- Sentences are tokenized and segmented by one of the segmentation algorithms described in the following section.
- Sentences longer than 150 tokens are removed.
- Sentences whose target to source ratio is below 0.7 are retained. This ratio is calculated based on the sentence lengths.
- Similar to Philip et al. (2019), a language match threshold is applied. Sentences rated 98% or higher are retained.
- Duplicate sentences are removed.

5 Methods

5.1 Segmentation

We compare two segmentation techniques: data-driven subwords and linguistically motivated subwords.

Subword segmentation refers to fully data-driven, non linguistically motivated segmentation algorithms (Sennrich et al., 2016c; Kudo and Richardson, 2018) that generate sub-words based on simpler frequency criteria to attain a predetermined vocabulary size. In our experiments we try out different vocabulary sizes as well as generating the subwords either individually for each language or jointly learning on both. The SentencePiece (**SP**) implementation (Kudo and Richardson, 2018) is used to perform this segmentation.

Linguistically Motivated Vocabulary Reduction (LMVR) is an unsupervised morphological segmentation algorithm based on Morfessor FlatCat (Kohonen et al., 2010; Grönroos et al., 2014) and proposed by Ataman et al. (2017). LMVR works by imposing an extra condition on the cost function of Morfessor so as to favour vocabularies of the desired size. When comparing regular Subword tokenization to LMVR, Ataman et al. (2017) report a +2.3 BLEU improvement on the English-Turkish translation task. Similar to SP, we need to set the vocabulary size prior to running the segmentation. LMVR models are trained separately for Tamil and English, which are then used to segment the respective datasets.

Name	Domain	EN Tokens(k)	TA Tokens(k)	Sentences(k)
Wikititles	Wikipedia	215	18	95
PMI	Political	707	87	40
UFAL	Mixed (News, Bible & Cinema)	3893	514	166
Koran	Religious	2366	586	92
MkB	Political (Speech)	104	15	6
PIB	Indian Press	1123	149	61
NLPC	Mixed	65	8	7
Wikimatrix	Mixed	2178	503	158
Total		10669	1885	625

Table 1: Approximate sizes (in thousands) of the Parallel Corpora used for training the NMT models

Name	Domain	TA Tokens(k)	Sentences(k)
Wikipedia Dumps	Wikipedia	4034	1669
News crawl	News	1496	709
PMI	Political	207	99
Total		5737	2477

Table 2: Approximate sizes (in thousands) of the Tamil Monolingual Corpora

5.2 Back-translation

In order to artificially increase the training data, we employ back-translation (BT) (Sennrich et al., 2016b). We consider two variations of this approach:

TaggedBT was presented by Caswell et al. (2019) and is similar to the original BT technique of Sennrich et al. (2016b), with the major difference being the addition of a special tag (here <BT>) in front of every back-translated English sentence. Caswell et al. (2019) had shown that this simple manoeuvre resulted in a higher BLEU score when compared to untagged BT based NMTs.

StupidBT Rather than performing actual BT which is expensive, Burlot and Yvon (2018) carry out the following:

1. Copy the target side data to the source side.

2. Prepend each token on the source side with a special id. For example, the token *tablet* becomes *bt_tablet*.

This simple and cost-effective technique was shown to perform almost on a par with regular BT on the English→French translation task.

5.3 Fine-tuning

Fine-tuning or transfer learning (Pan and Yang, 2010) is an effective technique to address a domain mismatch between the training set and the testset. While the testset consists of excerpts from newspapers, the training set consists of corpora with genres ranging from religious, political to movie subtitles. In fact, only a third of UFAL is news-oriented. A strategy to circumvent the domain mismatch is to fine-tune a pre-trained NMT system on a more domain specific dataset. Unfortunately the UFAL corpus is not domain tagged, so the news-only sentences cannot be easily retrieved.

We also excluded the PIB dataset due to its small size and large amount of almost identical sentences.

We hence perform fine tuning on the PMI dataset: This dataset consists of the sentences that were crawled from the Prime Minister of India's blog, with matters that are mostly political in nature. Despite the different content, we expect this corpus to be the closest in genre to the testset among the remaining parallel corpora.

5.4 Word Dropout

First introduced in Gal and Ghahramani (2016), the word dropout technique was modified by Sennrich et al. (2016a) to randomly drop tokens in-

stead of types during training. They reported an increase of 4-5 BLEU for the English↔Romanian language pair. Furthermore, Sennrich and Zhang (2019) report that introducing word dropout into NMT systems in low-resource settings leads to improvements in BLEU scores. We would hence like to investigate if the same improvements can be observed for EN-TA.

6 Experimental Setup

All our NMTs are developed using Fairseq (Ott et al., 2019). Following the architecture setup of Philip et al. (2019) the Transformer-Base implementation (BASE) is used, with slight changes to a few parameters, which are explained below. The encoder and decoder are both set to 5 layers with embedding dimension of 512 and 8 attention heads. The hidden layer dimension is 2048 and layer normalization is applied before each encoder and decoder layer. Other parameters were set as follows: dropout (0.001), weight decay (0.2) and batch size of 4k tokens. Our loss function is cross-entropy with label smoothing of 0.2. The model is trained for 100 epochs with early stopping criterion set to 3.

Segmentation The various segmentation algorithms are trained on the training data prior to the translation task. We report results with the following vocabulary sizes: 5k (source-target joint), 5k/5k, 10k/10k, 15k/15k and 20k/20k (source/target disjoint).

Back-Translation In order to perform BT, we first need to train a NMT model in the reverse direction, i.e. TA→EN. A Transformer based architecture is also used here. Our best configuration was: embedding and decoder having 6 layers, embedding layer having 512 dimensions and 6 attention heads with the rest of the parameters set as BASE. This model achieves a BLEU score of 18.27 on the UFAL TA-EN testset.

Fine-Tuning For the fine-tuning step, we take the pretrained BASE models and continue training them on the PMI dataset. An exhaustive search is done to find the best configurations for the fine tuning. The parameters with which we experimented are the learning rate, batch size, dropout and the value of label smoothing. Eventually we selected the following fine-tuning setup: learning rate of 0.002, batch size of 128, dropout of 0.3, label smoothing with factor of 0.3, and early stopping after 5 epochs without improvements.

Word Dropout Following Sennrich and Zhang (2019) we set the source word dropout to 0.3, i.e. the probability of a source word, in a batch, being dropped prior to training is 0.3.

7 Results

We report BLEU scores on three testsets: the UFAL testset (Ramasamy et al., 2012), half of the WMT2020 devset (DEV)[1] and the official WMT2020 testset. Given the rich morphology of Tamil, we also report CHRF scores (Popović, 2015) on the WMT2020 testset. We ran the program chrF++.py[2] with the arguments -nw 0 -b 3 to obtain the CHRF score.

From prior experimentation we found that a jointly trained SP model resulted in better BLEU when compared to separate training for each language, and hence perform the majority of SP experiments in Table 3 using a joint segmentation. On the other hand, LMVR being linguistically motivated is supposed to be trained independently for each language.

The last two contrastive experiments (Exp8.2 and Exp11.2) were run after the evaluation phase to better assess the impact of LMVR on translation quality in our best systems.

The following observations can be made based on the results:

Differences across testsets The trends are often inconsistent across testsets. Exp2 gave the highest BLEU score on UFAL (11.8) but a low BLEU score for DEV and WMT. On the other side, Exp11 (and Exp11.2) provided us the highest BLEU score on the official WMT testset, but a low 10.5 for UFAL. These variations could be attributed to the nature of the testsets and our training regime. Because we focused on improving our NMT systems to adapt to the news genre of WMT testset, this resulted in loss of translation accuracy of the UFAL testset, which was a mixture of three domains (one of them being news).

Effect of Back-translation Across both segmentation techniques, back-translation proved to be beneficial. Despite previously reported results, we found that fully fledged back-translation

[1] We randomly select one half of the WMT2020 devset for validation and use the remaining half for evaluation (DEV).

[2] https://github.com/m-popovic/chrF

	System	Segment.	Dict.size	BLEU UFAL	BLEU DEV	BLEU WMT	CHRF WMT
Exp1	BASE	SP	5k	11.2	8.5	5.1	42.8
Exp2	BASE +StupidBT	SP	5k	11.8	8.6	5.1	41.9
Exp3	BASE +TaggedBT	SP	5k	11.7	8.9	5.4	44.3
Exp6	BASE +TaggedBT	LMVR	5k/5k	11.1	9	5.6	40.1
Exp7	BASE +TaggedBT	LMVR	10k/10k	11.2	9.2	5.6	43.6
Exp8	BASE +TaggedBT	LMVR	15k/15k	11.1	9.3	6.0	48.1
Exp9	BASE +TaggedBT	LMVR	20k/20k	11.2	9.2	5.9	45.9
Exp11	BASE +TaggedBT+FT	LMVR	15k/15k	10.2	9.7	6.0	46.1
Exp13	BASE +TaggedBT+WD+FT	LMVR	15k/15k	10.7	**10.2**	**6.5**	**50.9**
Exp8.2	BASE +TaggedBT	SP	15k/15k	11.3	9.1	6.3	44.2
Exp11.2	BASE +TaggedBT+FT	SP	15k/15k	10.5	9.7	6.6	47.2

Table 3: English-Tamil results on three datasets: the general-domain UFAL (Ramasamy et al., 2012), our news development set (DEV) and the official WMT2020 news testset (WMT). Exp11 (in bold) was our official submission to WMT2020. SP refers to SentencePiece and LMVR to (Ataman et al., 2017). Dictionary size is given as one number for source-target joint segmentation, or as two numbers for source/target size when disjoint. FT and WD stand for fine-tuning and word dropout, respectively.

(TaggedBT) works considerably better than its cheaper approximation (StupidBT) on DEV, but not on the UFAL testset. While DEV reported increases of +0.3 (Exp2 vs. Exp3), a drop of -0.1 in BLEU was seen for UFAL. This could be due to the fact that Newscrawl was a major constituent of the monolingual corpora, that were used to train the TaggedBT systems. Also, when comparing a BASE system to one with TaggedBT (Exp1 vs. Exp3), we find an increase of +0.3 in BLEU. Given the DEV result, we decided to use fully fledged TaggedBT for the rest of our experiments.

SP vs. LMVR Based on our initial experiments, LMVR seemed to outperform SP. For instance, when comparing the TaggedBT systems with SP and LMVR (Exp3 vs. Exp9) we see a +0.5 increase in BLEU.

However, after the official submission, we performed additional contrastive experiments to account for LMVR having a much larger and disjoint vocabulary size (see Exp 8.2 vs. Exp8 and Exp11.2 vs. Exp11). In both settings, the linguistically motivated segmentation was actually outperformed by SentencePiece (+0.3 higher BLEU score on WMT). On the other hand, results were inconclusive when looking at the CHRF scores: namely, LMVR is much better than SP in the non fine-tuned system (Exp8 vs. Exp8.2), but slightly worse in the fine-tuned system (Exp11 vs. Exp11.2). These results seem to reveal a complex interplay between the effect of domain adaptation and the choice of an optimal segmentation strategy.

Effect of vocabulary size For our BT model with LMVR segmentation, we report the scores for four different vocabulary sizes (Exp6 to Exp9): among these, 15k for each language (Exp8) gives the best BLEU score of 9.3 on DEV. Therefore we use this size for the remaining experiments.

Effect of fine tuning When we compare models to their counterparts that were additionally fine-tuned, we observe a slight increase in the DEV BLEU score for the LMVR systems (compare Exp8 vs. Exp11) but unfortunately no effect on the WMT testset. This is probably due to the fact that the dataset on which we fine-tuned (PMI) was not close enough to the domain of the news translation testset.

Effect of word-dropout Word dropout was introduced to our best system, that is the one using TaggedBT and a LMVR vocabulary size of 15k/15k. The resulting system (Exp13) turned out to be our best performing system overall, but was not ready in time for the official submission. We find that the addition of word dropout resulted in a BLEU increase of +0.5 on DEV and WMT, and a large CHRF increase (+4.8) on WMT, which confirms the usefulness of this technique on a new lan-

Case	Case Marker	Tamil	SP	LMVR
Nominative	−∅	புத்தம் puththagam 'book'	புத்த+கம் puththa+gam	புத்த+கம் puththa+gam
Accusative	−அ −a	புத்தகம puththagama 'the book'	புத்தக+ம puththaga+ma	புத்த+க+ம puththa+ga+ma
Dative	−உக்கு −ukku	புத்தகமுக்கு puththagamukku 'to/for the book'	புத்தக+மு+க்கு puththaga+mu+kku	புத்த+கம+உக்கு puththa+gam+ukku
Genitive	−ஓட −ooda	புத்தகமோட puththagamooda 'the book's'	புத்தக+மோட puththaga+mooda	புத்த+க+மோ++ட puththa+ga+moo+da
Instrumental	−ஆல −aala	புத்தகமால puththagamaala 'by the book'	புத்தக+ம்+ஆல puththaga+m+aala	புத்த+க+மா++ல puththa+ga+maa+la
Sociative	−ஓட −ooda	புத்தகமோட puththagamooda 'along with the book'	புத்தக+மோட puththaga+mooda	புத்த+க+மோ++ட puththa+ga+moo+da
Locative	−ல −la	புத்தகம்ல puththagamla 'in the book'	புத்த+கம்+ல puththa+gam+la	புத்த+கம்+ல puththa+gam+la
Ablative	−லருந்து −larundhu	புத்தகம்லருந்து puththagamaala 'from the book'	புத்த+கம்+ல+ருந்து puththa+gam+la+rundhu	புத்த+கம்+ல+ருந்து puththa+gam+la+rundhu

Table 4: Different inflections of the Tamil singular noun புத்தகம். Columns SP and LMVR show the segmentations resulted by the SentencePiece (SP) and LMVR algorithms respectively.

guage pair.

8 Analysis

We also performed two small qualitative studies on the best systems based on segmentation. First, we compare how the segmentation algorithms segment the Tamil word புத்தகம் 'book'.

Secondly, using the example of the word *book* we observe how the systems translate the word to and from Tamil (Table 5).

Segmentation Table 4 shows how the word புத்தகம் and its various case forms are segmented by the segmentation techniques. The main differences that we observe are:

- LMVR and SP generated the same segmentation for three cases: nominative, locative and ablative.
- LMVR always generated segmentations with the base sub-word புத்த/puththa for all the case forms while SP generated the segments புத்த/puththa or புத்த/puththaga. This confirms the observations of Ataman et al. (2017), that LMVR produces more morphological segments.
- LMVR, on average, resulted in more segments per token than SP.

Translation Quality For the compound *comic-book*, the SP system translates it as நகைச்சுவை புக்/nakaiccuvai puk, i.e. *comedy book*, with the word புக்/puk being a direct transliteration for *book* and hence incorrect. On the other hand, LMVR provides the correct translation.

There were in total two occurrences of the nominative form of the புத்தகம்/puththagam, which were correctly translated by the two systems. The same was observed for the locative form புத்தகம்ல/puththagamla.

An example where both systems fail to translate the phrase *by the book* as in the sentence "I have every reason to believe they have done everything by the book and ...". The SP system provides

English	Tamil	SP	LMVR
comic-book	காமிக் புத்தக kaamik puththga	நகைச்சுவை புக் nakaiccuvai puk	காமிக் புத்தக kaamik puththga
book	புத்தகம் puththagam	புத்தகம் puththagam	புத்தகம் puththagam
in the book	புத்தகம்ல puththagamla	புத்தகம்ல puththagamla	புத்தகம்ல puththagamla
notebook	புத்தகத்தைத் puththagaththait	புத்தகட்டி puththagatti	குறிப்பேடு kurippetu
by the book	சட்டத்தைப் cattattaip	புத்தகமால் puththagamaal	விதிப்படி vithippati

Table 5: Qualitative Analysis of the Tamil word புத்தகம்/*puththagam* along with selected translations of the English word *book*

a grammatically correct form for book (ablative), it is however semantically incorrect. Meanwhile, the LMVR system generates the word விதிப்படி/vithippati meaning 'by rule' while the reference word has the meaning சட்டத்தைப்(bill).

Finally we observed with the word *notebook* that SP generated a non-existent word and LMVR provided an another translation for the English word *notebook*.

In the future, we aim to conduct in-depth analysis on what and which morphological features are captured by the NMT systems.

9 Conclusion

Although our results were not competitive with the other submissions for the EN-TA task, our paper presents the various settings that leads to an improvement in EN-TA translation. Mainly, we found that linguistically motivated subword segmentation (Ataman et al., 2017), which was previously shown to benefit translation from/into various non-Indian languages, does not consistently outperform the widely used SentencePiece segmentation despite the agglutinative nature of Tamil morphology. We also found that, for our English-Tamil systems, fully-fledged back-translation remains more competitive than its cheaper alternative (Burlot and Yvon, 2018). And finally, we observe a noticeable CHRF gain when adding word dropout (Sennrich et al., 2016a) to our best model.

Acknowledgements

Arianna Bisazza was funded by the Netherlands Organization for Scientific Research (NWO) under project number 639.021.646. We also would like to thank the Center for Information Technology of the University of Groningen for providing access to the Peregrine high performance computing cluster.

References

Duygu Ataman, Matteo Negri, Marco Turchi, and Marcello Federico. 2017. Linguistically motivated vocabulary reduction for neural machine translation from turkish to english. *The Prague Bulletin of Mathematical Linguistics*, 108(1):331 – 342.

Franck Burlot and François Yvon. 2018. Using monolingual data in neural machine translation: a systematic study. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 144–155, Brussels, Belgium. Association for Computational Linguistics.

Isaac Caswell, Ciprian Chelba, and David Grangier. 2019. Tagged back-translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 53–63, Florence, Italy. Association for Computational Linguistics.

Raj Dabre, Anoop Kunchukuttan, Atsushi Fujita, and Eiichiro Sumita. 2018. NICT's participation in WAT 2018: Approaches using multilingualism and recurrently stacked layers. In *Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation: 5th Workshop on Asian Translation: 5th Workshop on Asian Translation*, Hong Kong. Association for Computational Linguistics.

Yarin Gal and Zoubin Ghahramani. 2016. A theoretically grounded application of dropout in recurrent neural networks. In D. D. Lee, M. Sugiyama,

U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 1019–1027. Curran Associates, Inc.

Ulrich Germann. 2001. Building a statistical machine translation system from scratch: How much bang for the buck can we expect? In *Proceedings of the ACL 2001 Workshop on Data-Driven Methods in Machine Translation*.

Stig-Arne Grönroos, Sami Virpioja, Peter Smit, and Mikko Kurimo. 2014. Morfessor FlatCat: An HMM-based method for unsupervised and semi-supervised learning of morphology. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1177–1185, Dublin, Ireland.

Oskar Kohonen, Sami Virpioja, and Krista Lagus. 2010. Semi-supervised learning of concatenative morphology. In *Proceedings of the 11th Meeting of the ACL Special Interest Group on Computational Morphology and Phonology*, pages 78–86, Uppsala, Sweden. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Atul Kr. Ojha, Koel Dutta Chowdhury, Chao-Hong Liu, and Karan Saxena. 2018. The RGNLP machine translation systems for WAT 2018. In *Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation: 5th Workshop on Asian Translation: 5th Workshop on Asian Translation*, Hong Kong. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

S. J. Pan and Q. Yang. 2010. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359.

Jerin Philip, Shashank Siripragada, Upendra Kumar, Vinay Namboodiri, and C V Jawahar. 2019. Cvit's submissions to wat-2019. In *Proceedings of the 6th Workshop on Asian Translation*, pages 131–136, Hong Kong, China. Association for Computational Linguistics.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Loganathan Ramasamy, Ondřej Bojar, and Zdeněk Žabokrtský. 2012. Morphological processing for english-tamil statistical machine translation. In *Proceedings of the Workshop on Machine Translation and Parsing in Indian Languages (MTPIL-2012)*, pages 113–122.

Kengatharaiyer Sarveswaran, Gihan Dias, and Miriam Butt. 2019. Using meta-morph rules to develop morphological analysers: A case study concerning Tamil. In *Proceedings of the 14th International Conference on Finite-State Methods and Natural Language Processing*, pages 76–86, Dresden, Germany. Association for Computational Linguistics.

Sukanta Sen, Kamal Kumar Gupta, Asif Ekbal, and Pushpak Bhattacharyya. 2018. IITP-MT at WAT2018: Transformer-based multilingual indic-English neural machine translation system. In *Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation: 5th Workshop on Asian Translation: 5th Workshop on Asian Translation*, Hong Kong. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Edinburgh neural machine translation systems for WMT 16. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 371–376, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016c. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich and Biao Zhang. 2019. Revisiting low-resource neural machine translation: A case study. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 211–221, Florence, Italy. Association for Computational Linguistics.

The TALP-UPC System Description for WMT20 News Translation Task: Multilingual Adaptation for Low Resource MT

Carlos Escolano, Marta R. Costa-jussà, José A. R. Fonollosa,
{carlos.escolano,marta.ruiz,jose.fonollosa}@upc.edu,
TALP Research Center
Universitat Politècnica de Catalunya, Barcelona

Abstract

In this article, we describe the TALP-UPC participation in the WMT20 news translation shared task for Tamil-English. Given the low amount of parallel training data, we resort to adapt the task to a multilingual system to benefit from the positive transfer from high resource languages. We use iterative back-translation to fine-tune the system and benefit from the monolingual data available. In order to measure the effectivity of such methods, we compare our results to a bilingual baseline system.

1 Introduction

Modern NMT systems such as Transformer require large amounts of training data in order to obtain good generation results. For this reason, low resource languages represent a good opportunity to explore new techniques to treat data more efficiently and benefit from available sources of data like monolingual corpora.

From the WMT20 news tasks proposed languages we are presenting our results on the English-Tamil language pair, Tamil is an official language from India, Sri Lanka, and Singapore having approximately 75 million native speakers. It belongs to the Dravidian family, originated in Asia.

Two principal reasons can make Tamil a challenging language for machine translation: script and agglutination. Tamil's script consists of 12 vowels and 18 consonants plus one special character, allowing the combination of 247 possible characters. Compared to the Latin script employed by most western languages, it is an order of magnitude higher in the number of possible characters.

Also, by agglutination, suffixes can be added to root words to form new ones. These words can lead to multiple words in the target language in the context of machine translation, which may affect attention and decoding in NMT systems.

This work discusses the system proposed for the evaluation in which we combine the use of multilingual parallel data with monolingual data to boost the performance of our proposed NMT system.

2 Low Resource NMT

Modern NMT systems benefit from having hundreds of thousands or even millions of parallel sentences. When working with low resource language pairs, the two main approaches are the use of monolingual corpora and multilingual NMT. While parallel data may be difficult to obtain for low resource languages, monolingual data is usually more available, as it does not require any additional labeling.

A common approach to benefit from monolingual data is back-translation (Sennrich et al., 2016a), which consists of translating a monolingual corpus to generate synthetic corpora that can be later employed to continue training. Similar techniques create a synthetic pseudo-parallel corpus through a pivot language (Casas et al., 2019) that can be then trained similarly to back-translation when data is available between the desired language pair and a pivot high resource language. More recently, iterative back-translation (Hoang et al., 2018) was proposed. This technique allows the system to generate synthetic data while updating the system, so better the new data improves as the system trains. On the other hand, several works on Multilingual NMT have shown benefits for low resource language pairs by allowing positive transfer from the high resource languages, boosting the performance of the low resource ones. Different architectures have been proposed that show this behavior, from universal models where all parameters are shared between all languages (Johnson et al., 2017), to architectures that share a common device that maps representations into a shared represen-

Proceedings of the 5th Conference on Machine Translation (WMT), pages 134–138
Online, November 19–20, 2020.

tation space (Firat et al., 2016; Zhu et al., 2020), to architectures that do not share parameters (Escolano et al., 2019; Escolano et al.; Schwenk and Douze, 2017).

In the context of the WMT20 Tamil-English news shared task, as the provided parallel data is limited, we resorted to a combination of both proposed methods by incrementally train the new language pair into a Multilingual NMT system using the provided parallel data, to later fine-tune the system using iterative-back-translation with monolingual corpora.

3 Related Work

Previous works (Choudhary et al., 2018) have shown that Indian languages are usually a challenge for NMT systems due to their difference in terms of vocabulary and grammar compared to western languages such as English. Also, standard preprocessing methods do not always work well with them, so specific solutions are required to obtain good results.

In the context of NMT, previous systems, such as MIDAS (Choudhary et al., 2018), proved that the use of subword units leads to significant improvements in translation quality when applied to Tamil by preventing Out of Vocabulary words in at generation time.

4 Corpora and Data Preparation

All proposed systems in this work are constrained using exclusively data provided by the task's organization. The multilingual initial system was trained using *Europarl v8*, for all translation directions between English, French, Spanish, and German. For English-Tamil *PMIndia*, *Tanzil v1*, *The UFAL EnTam corpus*, The *NLPC UOM En-Ta corpus*, *Wikimatrix*, and *Wikitiles*. As monolingual Tamil data, we used News Crawl, while for English, we used *News-commentary*.

We processed all non-Tamil data following *Moses* (Koehn et al., 2007) scripts provided by the organization. For each one, we applied punctuation normalization, tokenization, and true-casing. Then each language is independently tokenized using BPE (Sennrich et al., 2016b) with 32 thousand operations. Table 1 the estatistics for each language. Tamil data has been tokenized at word-level using *Indic-NLP* (Kunchukuttan, 2020) and then tokenized with BPE with 16 thousand operations.

corpus	lang	sentences	words
DE-EN	DE	1758872	40265543
	EN	1758872	40265543
DE-ES	DE	1663458	37698204
	ES	1663458	40808518
DE-FR	DE	1681466	37410662
	FR	1681466	43056346
EN-ES	EN	1769606	41803882
	ES	1769606	43156309
EN-FR	EN	1770112	41211543
	FR	1770112	45196313

Table 1: Corpus statistics in number of words and sentences for the language pairs of the Multilingual initial system.

corpus	lang	set	sentences	words
EN-TA	EN	train	494310	7355160
		test	1275	29774
	TA	train	494310	15163570
		test	1275	66564
EN	EN	train	608912	14995557
TA	TA	train	504320	6426186

Table 2: Corpus statistics in number of words and sentences for the English-Tamil parallel data and English and Tamil monolingual training sets.

Table 2 show the statistics for the parallel Englist-Tamil data as well as the monolingual corpora used.

As test set, we used 1275 lines extracted from the development set provided from the organization, keeping the remaining ones as validation set.

5 System Description

In this section, we are going to discuss the details of the pipeline followed to create the translations systems for this submission, including the multilingual supervised pretraining and the unsupervised fine-tuning using monolingual corpora.

5.1 Multilingual Supervised Pretraining

Methodology. Following the proposed model in (Escolano et al., 2020), new languages can be added to the system without retraining the system, just using parallel data to one of the initial ones. In this work, we added Tamil using the provided parallel data to English. To train the new Tamil to English translation direction, a new Tamil encoder is added to the system with the previous English encoder frozen, to prevent the model from affecting the performance of the remaining pairs. Training with the

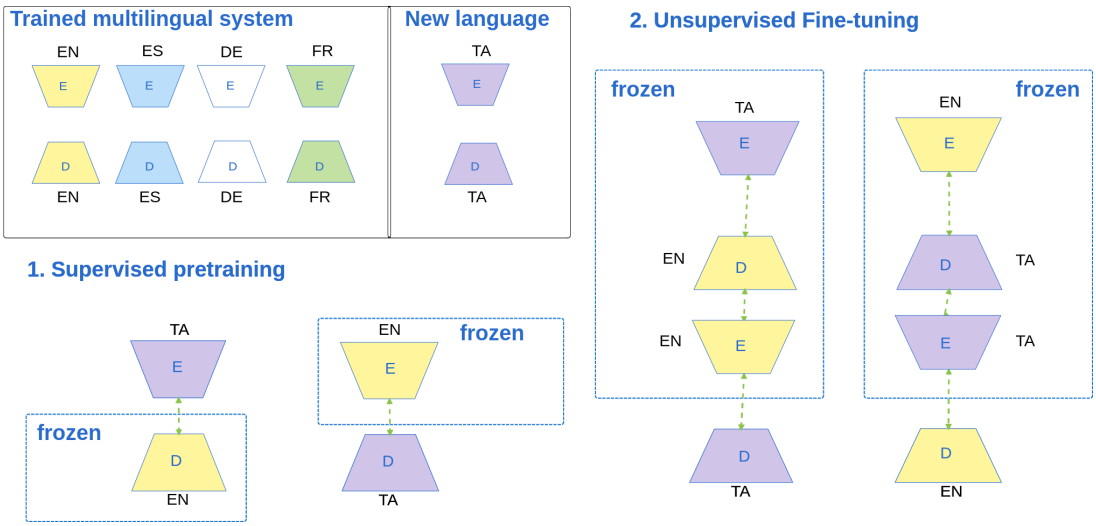


Figure 1: Training pipeline. Step 1 Supervised preptraining, Step 2 Unsupervised fine-tuning.

frozen decoder induces the new encoder to learn a similar representation to the ones already in the multilingual model. In addition, as the English decoder has been trained with more data from all the language pairs in the Multilingual NMT system, we have positive transfer from the frozen modules to the new ones, boosting the translation performance compared to the bilingual NMT baseline. Following the same principles, the English-Tamil translation direction is trained by freezing the English encoder and training the Tamil decoder to force the shared representation. In this case, we also notice the positive transfer compared to the baseline trained with just parallel data. See in Figure 1 the schema of the supervised pretraining that we have just described.

Implementation. For this work, all encoders and decoders were implemented using the Transformer (Vaswani et al., 2017) architecture, with 6 layers, 8 heads, 512 embedding size, and 2048 feed-forward size for each of them, and everything was implemented using *Fairseq*'s(Ott et al., 2019) 0.6 release. The multilingual NMT model was trained in a single NVIDIA TITAN XP for 50 thousand updates using adam optimizer with 0.001 as learning, 4000 warmup updates and updating every 16 batches of 2000 tokens. Adding Tamil-English and English-Tamil directions to the system took approximately 45 thousand updates using the same parameters and GPU configuration.

5.2 Monolingual Unsupervised Fine-tuning

Methodology. The previous process has benefited from the additional corpus from the Multilingual NMT system, but as stated before, monolingual data is another common source of improvement for NMT systems. In this section, we are going to discuss how we added monolingual data to the previously described model. To employ the monolingual data available in our system, we define an autoencoder using the already trained encoder and decoder modules in the given language. These modules are not trained to regenerate the input, we introduce an adaptor, between both modules, responsible for processing the representation generated and output a new one understood by the decoder. Taking advantage of the architecture, we can use one of the decoders to greedy decode the representation created and encode it back with one the encoders, to compute the reconstruction of the monolingual input. Figure 1 showcases in "unsupervised fine-tunning" how is process is applied in our work to use both Tamil monolingual data with an English adaptor and English data with the Tamil adaptor.

In this work, both encoder and adaptor were frozen, and only the final decoder was updated. As future work, then encoder could be also trained, improving the representations generated at each training epoch.

Implementation. As the rest of the architecture, this process has been implemented using the same GPU and parameter configuration, in this case for approximately 6 thousand updates.

5.3 Post-processing

Once our model is fully trained the apply an additional step of checkpoint averaging in which the n checkpoints containing the weights of the network are combined using the defaults script provided by *Fairseq*.

In this work, given that the corpus was small we saved checkpoint every epoch of approximately 400 updates and averaged the last 4 checkpoints saved.

Finally, to generate the final submissions, detruecasing and detokenization using the scripts provided by Moses to the English outputs, while IndicNLP detokenization is applied to the Tamil ones.

6 Experiments and Results

The motivation for this work was to explore the combination of both positive transfer and monolingual data in a low resource task such as English-Tamil Translation.

To test our hypothesis we trained a bilingual baseline with just the parallel data available for the task and compared its results to an incremental using adaptation to a multilingual NMT system and monolingual fine-tuning to measure the impact of each measure in the final performance. All configurations have the same architecture and number of parameters and have been tested on the same 1275 lines extracted from the *newsdev2020* Tamil-English set.

To introduce some context about the multilingual system, we evaluated its performance using *newstest13* as test set, and the performance English performance ranged from 20.31 BLEU points from the English-German translations direction, to 29.74 for English-French. When English is the target language the results vary from 24.54 for German-English, to 27.75 for Spanish-English. About the impact of positive transfer from Multilingual NMT, Tables 4 and 3 show that both directions benefit from adding Tamil into the MNMT system with improvement of 1.58 and 4.09 BLEU points respectively, approximately a 40% better than the bilingual baseline in both directions.

When looking at the monolingual fine-tuning results, we can observe that the English to Tamil translation direction benefits more (2.65 BLEU) from the technique than the Tamil to English direction (1.02 BLEU). This difference in the performance may be explained by the difference in the training of both decoders. While the Tamil decoder has been trained just with the parallel data for the task, the English decoder was trained with the multilingual NMT system with more data available, which may lead to a more robust model to fine-tuning.

Finally, looking at the checkpoint averaging results, in both directions it leads to a small improvement, less than 0.2 BLEU, showing limited impact in the final results.

System	BLEU	ΔBLEU
Baseline	3.42	-
Multilingual	5.00	1.58
+ Mono	7.65	2.65
+ Checkpoint Avg	7.92	0.27

Table 3: Results measured in BLEU of the English to Tamil Translation direction.

System	BLEU	ΔBLEU
Baseline	6.51	-
Multilingual	10.6	4.09
+ Mono	11.62	1.02
+ Checkpoint Avg	11.8	0.18

Table 4: Results measured in BLEU of the Tamil to English Translation direction.

7 Conclusions

In this paper, we described the TALP-UPC participation in the WMT20 news translation shared task for Tamil-English. The motivation of this work was to explore the combination of multilingual transfer from high resource languages and monolingual data applied to low resource NMT. Our experiments showcase the effectiveness of adapting low resource languages pre-trained multilingual systems and how it introduces positive transfer compared to a bilingual baseline system. Also it shows that monolingual data can be successfully introduced into the system and that it can boost the performance of the system. As future work, we could explore the fine-tuning of both encoder and decoder during the monolingual unsupervised fine-tuning in order to help the system produce better synthetic data as the training takes place.

Acknowledgments

This work is supported in part by the Google Faculty Research Award 2019, the Spanish Ministerio de Ciencia e Innovación, through the postdoc-

toral senior grant Ramón y Cajal and by the Agencia Estatal de Investigación through the projects EUR2019-103819, PCIN-2017-079 and PID2019-107579RB-I00 / AEI / 10.13039/501100011033

References

Noe Casas, José A. R. Fonollosa, Carlos Escolano, Christine Basta, and Marta R. Costa-jussà. 2019. The TALP-UPC machine translation systems for WMT19 news translation task: Pivoting techniques for low resource MT. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 155–162, Florence, Italy. Association for Computational Linguistics.

Himanshu Choudhary, Aditya Kumar Pathak, Rajiv Ratan Saha, and Ponnurangam Kumaraguru. 2018. Neural machine translation for English-Tamil. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 770–775, Belgium, Brussels. Association for Computational Linguistics.

Carlos Escolano, Marta R. Costa-jussà, and José A. R. Fonollosa. 2019. From bilingual to multilingual neural machine translation by incremental training. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 236–242, Florence, Italy. Association for Computational Linguistics.

Carlos Escolano, Marta R Costa-jussà, José AR Fonollosa, and Mikel Artetxe. 2020. Multilingual machine translation: Closing the gap between shared and language-specific encoder-decoders. *arXiv preprint arXiv:2004.06575*.

Carlos Escolano, Marta R. Costa-Jussà, and José A. R. Fonollosa. From bilingual to multilingual neural-based machine translation by incremental training. *Journal of the Association for Information Science and Technology*, n/a(n/a).

Orhan Firat, Baskaran Sankaran, Yaser Al-Onaizan, Fatos T. Yarman Vural, and Kyunghyun Cho. 2016. Zero-resource translation with multi-lingual neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 268–277, Austin, Texas.

Vu Cong Duy Hoang, Philipp Koehn, Gholamreza Haffari, and Trevor Cohn. 2018. Iterative back-translation for neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 18–24, Melbourne, Australia. Association for Computational Linguistics.

Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the ACL: Demo Papers*, pages 177–180.

Anoop Kunchukuttan. 2020. The IndicNLP Library. https://github.com/anoopkunchukuttan/indic_nlp_library/blob/master/docs/indicnlp.pdf.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Holger Schwenk and Matthijs Douze. 2017. Learning joint multilingual sentence representations with neural machine translation. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 157–167, Vancouver, Canada. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Changfeng Zhu, Heng Yu, Shanbo Cheng, and Weihua Luo. 2020. Language-aware interlingua for multilingual neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1650–1655, Online. Association for Computational Linguistics.

An Iterative Knowledge Transfer NMT System for WMT20 News Translation Task

Jiwan Kim, Soyoon Park, Sangha Kim, Yoonjung Choi
Individual Researchers
{jiwan.kim37,parksso0223,shkim000,choiyj35}@gmail.com

Abstract

This paper describes our submission to the WMT20 News translation shared task in English to Japanese direction. Our main approach is based on transferring knowledge of domain knowledge and linguistic characteristics by pre-training the encoder-decoder model with large amount of in-domain monolingual data through unsupervised and supervised prediction task. We then fine-tune the model with parallel data and in-domain synthetic data which is generated by iterative back-translation. For additional gain, we generate final results with an ensemble model and re-rank them with averaged models and language models. Through these methods, we achieve +5.42 BLEU score compared to the baseline model.

1 Introduction

This paper describes our submission to the WMT20 News translation task in English to Japanese direction. In this year, English-Japanese directions have newly established in News Translation Shared Task. The English-Japanese translation is not easy to deal with because of the difference in word order and the rich morphological characteristics of Japanese. Nevertheless, recent architectures for Neural Machine Translation (NMT), such as Transformer (Vaswani et al., 2017), show reasonable results when we have enough parallel data. Unfortunately, however, there is not much in-domain parallel data provided for English-Japanese task. To solve this issue, in this paper, we suggest the iterative knowledge transfer system which pre-trains the model with in-domain monolingual data.

Our system is based on Transformer architecture. We pre-train the model to transfer linguistic characteristics and domain knowledge of monolingual data. Although there are various pre-training methods for NMT, MASS (Song et al., 2019) is adopted in our system since MASS pre-trains the encoder and the decoder jointly and uses both labeled data and unlabeled data as the training data. To supplement insufficient in-domain parallel data, we generate synthetic data by back-translation from in-domain monolingual data. We also add some noise to the synthetic data. We then pre-train the model with the synthetic parallel data for supervised method and the monolingual data for unsupervised way. In fine-tuning step, we train the model with parallel corpus and perform the back-translation with in-domain data for iterative fine-tuning. In addition, we adopt an ensemble and averaging methods which are simple but very effective to improve performance in deep learning. With ensemble and average models, we apply noisy channel re-ranking which shows higher performance compared to R2L re-ranking (Yee et al., 2019). Through these methods, we achieve +5.42 BLEU score (Papineni et al., 2002; Post, 2018) compared to the baseline model.

2 Approach

Our system aims to encourage knowledge extraction of domain knowledge and linguistic characteristics by iteratively performing pre-training and fine-tuning. In this section, we explain techniques we use in each step.

2.1 Pre-training strategy

MASS is a masked sequence to sequence pre-training method for the encoder-decoder based language generation tasks (Song et al., 2019). The advantage of MASS is that it uses the encoder-decoder framework to predict the masked part given the masked sentence. Several consecutive tokens in a sentence are randomly masked; the encoder takes them as input, and the decoder is trained to predict masked tokens. This method allows MASS to learn the capability of representation

Proceedings of the 5th Conference on Machine Translation (WMT), pages 139–144

extraction. In this paper, we adopt both supervised and unsupervised prediction methods of MASS. There are plenty of in-domain monolingual corpus but insufficient in-domain parallel corpus. Thus, we generate synthetic data by back-translation and apply supervised prediction task. In addition, we use large amount of out-domain monolingual corpus for unsupervised prediction task to encourage the ability of language modeling.

Let $x \in \mathcal{X}$ as an monolingual source sentence, and m is the number of tokens of sentence x. We denote $x^{\setminus u:v}$ as an modified sentence of x where its position u to v are masked, $0 < u < v < m$. $x^{u:v}$ denotes the original sentence fragment of x from u to v. Those sentences can have different fragment positions u and v for each. In the sentence fragment, we replace each masked token to a special symbol $[\mathbb{M}]$, so the number of words in the sentence is not changed. Then, we train model with the masked sentence $x^{\setminus u:v}$ to predict the sentence fragment $x^{u:v}$. Supervised setting is used also where bilingual sentence pair $(x, y) \in (\mathcal{X}, \mathcal{Y})$ can be leveraged for pre-training. It is trained to predict y from the input $x^{\setminus u:v}$. The log likelihood in the entire setting is as follows:

$$
\begin{aligned}
L(\theta; (\mathcal{X}, \mathcal{Y})) &= \frac{1}{|\mathcal{Y}|} \sum_{(x,y)\in(\mathcal{X},\mathcal{Y})} \log P(y|x^{\setminus u:v}; \theta) \\
&+ \frac{1}{|\mathcal{X}|} \sum_{(x,y)\in(\mathcal{X},\mathcal{Y})} \log P(x|y^{\setminus u:v}; \theta) \\
&+ \frac{1}{|\mathcal{X}|} \sum_{x\in\mathcal{X}} \log P(x^{u:v}|x^{\setminus u:v}; \theta) \\
&+ \frac{1}{|\mathcal{Y}|} \sum_{y\in\mathcal{Y}} \log P(y^{u:v}|y^{\setminus u:v}; \theta)
\end{aligned} \tag{1}
$$

$P(y|x^{\setminus u:v}; \theta)$ and $P(x|y^{\setminus u:v}; \theta)$ denote the probability of translating a masked sequence to another language. This prediction task encourages the encoder to extract meaningful representations of masked input tokens in order to predict the unmasked output sequence.

2.2 Noised back-translation

Inspired from the noised back-translation (Edunov et al., 2018; Wu et al., 2019), we add noise to the train corpus. Let X and Y denote two languages, and let $\mathcal{X}$ and $\mathcal{Y}$ denote two corresponding sentence corpora, a set of all sentences. Let $\mathcal{B} = \{(x_i, y_i)_{i=1}^{N}\}$ denote the bilingual training corpus, where $x_i \in \mathcal{X}$, $y_i \in \mathcal{Y}$, and N is the number of sentence pairs. Let $\mathcal{M}_x = \{x_j\}_{j=1}^{N_x}$ and $\mathcal{M}_y = \{y_j\}_{j=1}^{N_y}$ denote sets of monolingual sentences, where N_x and N_y are sizes of each set, $x_j \in \mathcal{X}, y_j \in \mathcal{Y}$. We then train models $f_b : \mathcal{X} \mapsto \mathcal{Y}$ and $g_b : \mathcal{Y} \mapsto \mathcal{X}$ on the given bilingual data $\mathcal{B}$. Then, we build the following two synthetic datasets through the trained models:

$$
\begin{aligned}
\bar{\mathcal{B}}_{sx} &= \{(x, f_b(x))|x \in \mathcal{M}_x\}, \\
\bar{\mathcal{B}}_{sy} &= \{(y, g_b(y))|y \in \mathcal{M}_y\}, \\
\bar{\mathcal{B}}_{tx} &= \{(f_b(x), x)|x \in \mathcal{M}_x\}, \\
\bar{\mathcal{B}}_{ty} &= \{(g_b(y), y)|y \in \mathcal{M}_y\}
\end{aligned} \tag{2}
$$

where $\bar{\mathcal{B}}_{sx}$, $\bar{\mathcal{B}}_{sy}$ can be seen the forward translation of source-side monolingual data of X and Y and $\bar{\mathcal{B}}_{tx}$, $\bar{\mathcal{B}}_{ty}$ can be seen the backward translation of target-side monolingual data of X and Y.

We build following noise versions of the augmented datasets for training.

$$
\begin{aligned}
\bar{\mathcal{B}}^n_x &= \{(\sigma(x), \sigma(y))|(x, y) \in (\bar{\mathcal{B}}_{sx} \cup \bar{\mathcal{B}}_{ty})\}, \\
\bar{\mathcal{B}}^n_y &= \{(\sigma(y), \sigma(x))|(y, x) \in (\bar{\mathcal{B}}_{sy} \cup \bar{\mathcal{B}}_{tx})\}
\end{aligned} \tag{3}
$$

where $\sigma(x)$ denote the noised sentence of x, which consists of two types of noise: deleting tokens with probability 0.05 and swapping tokens in the sentence, implemented as a random permutation over the tokens with the uniform distribution but restricted to swapping words no further than three positions apart, where three is set empirically.

2.3 Noisy channel re-ranking

Noisy channel re-ranking method (Yee et al., 2019) is derived from Bayes' rule.

$$
p(y|x) = \frac{p(x|y)p(y)}{p(x)} \tag{4}
$$

Let x as a source sequence and y as a target sequence. Since $p(x)$ is constant for all y, only the channel model $p(x|y)$ and the language model $p(y)$ determine y when x is given. Score used for re-ranking can be calculated as follows:

$$
\frac{\alpha * logp(y|x) + \beta * logp(x|y) + \gamma * logp(y)}{|y|^p} \tag{5}
$$

where α, β, γ are tunable weight, and p is length penalty for target length $|y|$.

3 Experiments

3.1 Data

Data statistics The training data of the entire system is shown in Table 1. We use News Commentary (NC) data as another validation set in addition to newsdev2020 (devset).

Dataset	Lines
Parallel Data	
Wiki_Titles v2	0.7M
WikiMatrix	3.89M
Japanese-English Subtitle Corpus	2.8M
The Kyoto Free Translation Task	0.44M
TED Talks	0.24M
Monolingual Data (En)	
Europarl v10	2.29M
News Commentary v15	0.6M
News Crawl	23.35M
News Discussions	63.51M
Monolingual Data (Ja)	
News Crawl	3.44M
News Commentary v15	2983
Common Crawl	1773.97M

Table 1: Training corpora for our system

Preprocessing We use recaser in Moses (Koehn et al., 2007) to recase Japanese-English Subtitle Corpus where English side is lowercased. We also normalize punctuation marks and tokenize English corpus with Moses. We use Mecab (Kudo, 2006) to tokenize Japanese corpus. We adopt Sentencepiece (Kudo and Richardson, 2018); separate vocabs with 32K tokens are generated for each language. Separate vocabs show higher score in BLEU than a joint vocab in English-Japanese.

Filtering We first filter the parallel corpus based on length; sentences with more than 800 characters are removed from the training data. We then filter the training corpus with LangId (Lui and Baldwin, 2012). If LangIds of source or target side are mismatched, we filter out this data.

Data selection Unlike English, there are not enough news data in Japanese, so we select data from Common Crawl and use them as in-domain data. To obtain data close to in-domain, we classify sentences into in-domain and out-domain based on the perplexity of in-domain and out-domain language model (Moore and Lewis, 2010).

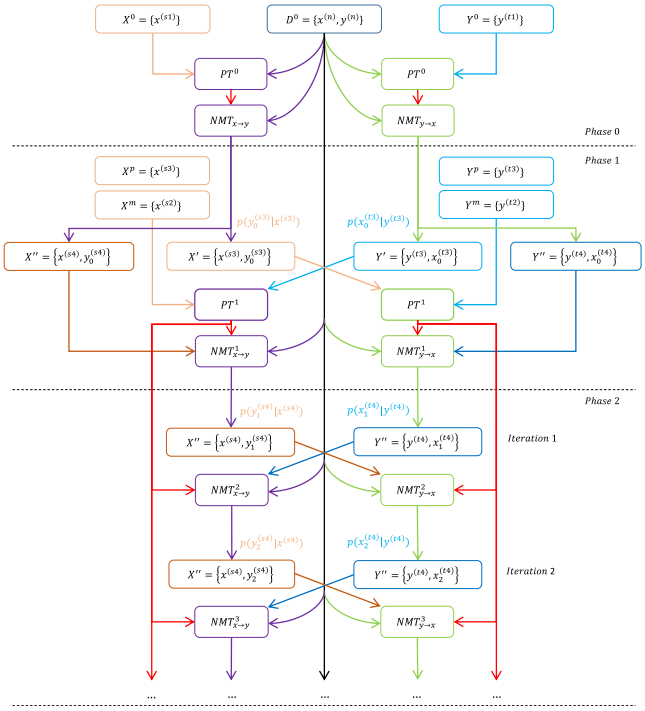


Figure 1: Illustration of training sequences of our system, where pre-trained models PT^* on both side are identical but separated for clarity.

Let $PPL_{in}(s)$ as the perplexity for sequence s with the in-domain language model and $PPL_{out}(s)$ as same with the out-domain language model. To classify sentences as close to in-domain, We calculate a score as follows:

$$\mathcal{S} = PPL_{out}(s) - PPL_{in}(s) \quad (6)$$

We train in-domain and out-domain language models respectively with KenLM (Heafield, 2011). The in-domain language model is trained with News Crawl corpus and the out-domain language model is trained with Common Crawl corpus.

3.2 Experimental setting

Our system is based on Transformer-big model on Fairseq (Ott et al., 2019)[1], which consists of 6-layers encoder and decoder each with 1024 embedding & hidden size and 4096 feed-forward layer size. Our system is trained using MASS[2] on 16×V100 GPUs, both in pre-training and fine-tuning.

3.3 Pre-training

Our entire training sequence is described in Figure 1. For the phase 0, we randomly sample 10M

[1] https://github.com/pytorch/fairseq
[2] https://github.com/microsoft/MASS

sentences X^0 and Y^0 from each mono corpus for unsupervised prediction task and use all available parallel corpus D^0 for supervised task. We prepare two separated prediction tasks, supervised and unsupervised setups respectively. For the supervised setup, we randomly mask entire input tokens in each sentence by 30% probability. In the unsupervised setup, we mask the fragment by replacing consecutive tokens with symbol $[\mathbb{M}]$ from random start position u. It first chooses 30% from input tokens, and each i-th token will be replaced as (1) an unchanged i-th token by 80% of the time, (2) a random token by 10% of the time, and (3) a masked token $[\mathbb{M}]$ by 10% of the time. After pre-training of model PT^0, two fine-tuned models $NMT_{x\to y}$ and $NMT_{y\to x}$ are trained with the parallel corpus, English-Japanese and Japanese-English direction respectively.

Lang	Lines	Remark
en	20M	
ja	20M	
ja*-en	5M	Randomly filtered
en*-ja	5M	LM-based filtered

Table 2: An amount of training corpora for pre-training. * means back-translated data from correspond monolingual corpus.

In the beginning of next phase, we create a new setup and train the model with training data mentioned in Table 2. We add noised synthetic data X' and Y' to create following version of training data. It consists of $\bar{\mathcal{B}}_{sx}$, $\bar{\mathcal{B}}_{sy}$, $\bar{\mathcal{B}}^n_x$ and $\bar{\mathcal{B}}^n_y$. X^m and Y^m consist of 20M mono corpora for unsupervised pre-training. 5M English mono corpus are randomly chosen from mono corpus, and 5M Japanese mono corpus are selected based on Equation 6; they are represented as X^p and Y^p in Figure 1. Then, 5M mono corpora are translated with $NMT_{x\to y}$ and $NMT_{y\to x}$ respectively.

PT^1 model is trained with above train corpus. Then, we train two fine-tuned models, $NMT^1_{x\to y}$ and $NMT^1_{y\to x}$ separately with parallel corpora in Table 3.

3.4 Iterative fine-tuning

After pre-training in phase 1, we create fine-tuned models with parallel corpus D^0 and synthetic corpus X'' and Y''.

Inspired from joint training (Zhang et al., 2018), we perform back-translation and fine-tune steps iteratively in phase 2. Synthetic corpora for each steps are replaced to a newly generated ones from developed models, which are represented as X'' and Y'' in Figure 1.

Lang	Lines	Domain	Remark
			English - Japanese
en-ja	7M	out	
en*-ja	3M	in	
			Japanese - English
ja-en	7M	out	
ja*-en	7M	in	Randomly filtered

Table 3: An amount of training corpora for fine-tuning

3.5 Advance decoding

We improve our final result with noisy channel re-ranking method (Yee et al., 2019). The small difference is we use the different direct model for scoring instead of using the same model used for generation. To generate y, we first ensemble three models with final back-translated models, considering validation sets. We generate 44 n-bests results with 44 beam size with ensemble models. Then, we re-rank the results according to Equation 5. The direct model for scoring is the averaged model of three models used for ensemble. This is faster and shows better results compared to the ensemble model. The channel model is an average model in the opposite direction. For language model, we use Transformer-big model, trained only with News domain monolingual corpus. Finally, we tune weights of each model and length penalty with validation sets.

3.6 Experimental Results

Step	Model	Dev
	Baseline	17.62
Phase 0	MASS	19.16
Phase 1	MASS	19.23
	+ Noise	19.31
Phase 2	Back-translation Iter1	23.59
	Back-translation Iter2	23.91
	Ensemble	24.05
	+ Beam 44	24.21
	Re-ranking(devset)	24.73
	Re-ranking(NC)	23.55

Table 4: En-Ja BLEU scores on WMT20 devset

Model	Test
Baseline	20.51
Ensemble + Beam 44	25.05
Re-ranking(devset)	24.41
Re-ranking(NC)	**25.93**

Table 5: En-Ja BLEU scores on WMT20 test set.

The results of English to Japanese direction are shown in Table 4 and 5. Our final submission's BLEU score is 5.42 higher than the baseline model.

For evaluation, multi-bleu.perl[3] is used after tokenizing with Mecab in Japanese. The baseline model is trained only with parallel data in Transformer-big architecture and is decoded with beam size 4. It shows great performance improvement when MASS is applied. When using synthetic data and adding noise to data in pre-training steps (Phase 1), it shows better results compared to it with only parallel data (Phase 0). Back-translation with the in-domain monolingual data increases the BLEU score most, and the score increases further in the next iteration. The ensemble model and large beam size also show better BLEU score.

For the test set, we replace symbol £ to "pound" in source sentences as pre-processing. We re-rank and tune the parameters based on News Commentary parallel data set which shows better results than tuning with devset. Since we select best models based on devset in previous steps, using devset in re-ranking seems to result in overfitting.

The final result of our submission is shown in Table 6. Characters based tokenizer and SacreBLEU[4] are used for evaluation in Ocelot.

Submission	SacreBLEU	chrF
English-Japanese	41.0	0.351

Table 6: Automatic evaluation on WMT20 test set in Ocelot.

4 Conclusions

In this paper, we describe our submission to the WMT20 news translation task in English to Japanese direction. Our main approach is based on transferring knowledge from large amount of monolingual data by pre-training the model iteratively using MASS. We then improve the system with several effective methods: noised and iterative back-translation, in-domain data selection, and re-ranking. Through these methods, we achieve competitive results compared to the baseline and prove that the iterative knowledge transfer system we proposed is effective.

[3] https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/multi-bleu.perl

[4] https://github.com/mjpost/sacrebleu

References

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. *CoRR*, abs/1808.09381.

Kenneth Heafield. 2011. Kenlm: Faster and smaller language model queries. In *Proceedings of the sixth workshop on statistical machine translation*, pages 187–197.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Taku Kudo. 2006. Mecab: Yet another part-of-speech and morphological analyzer. *http://mecab. sourceforge. jp*.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226*.

Marco Lui and Timothy Baldwin. 2012. langid. py: An off-the-shelf language identification tool. In *Proceedings of the ACL 2012 system demonstrations*, pages 25–30.

Robert C. Moore and William Lewis. 2010. Intelligent selection of language model training data. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 220–224, Uppsala, Sweden. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting bleu scores. In *Proceedings of the Third Conference on Machine Translation, Volume 1: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. MASS: masked sequence to sequence pre-training for language generation. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 5926–5936. PMLR.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Lijun Wu, Yiren Wang, Yingce Xia, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. 2019. Exploiting monolingual data at scale for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4207–4216, Hong Kong, China. Association for Computational Linguistics.

Kyra Yee, Yann Dauphin, and Michael Auli. 2019. Simple and effective noisy channel modeling for neural machine translation. In *Conference on Empirical Methods in Natural Language Processing*.

Zhirui Zhang, Shujie Liu, Mu Li, Ming Zhou, and Enhong Chen. 2018. Joint training for neural machine translation models with monolingual data. *CoRR*, abs/1803.00353.

Tohoku-AIP-NTT at WMT 2020 News Translation Task

Shun Kiyono[♠♢] **Takumi Ito**[♢♠] **Ryuto Konno**[♢♠] **Makoto Morishita**[♡♢] **Jun Suzuki**[♢♠*]
[♠]RIKEN Center for Advanced Intelligence Project [♢]Tohoku University
[♡]NTT Communication Science Laboratories
shun.kiyono@riken.jp;
{t-ito, ryuto, jun.suzuki}@ecei.tohoku.ac.jp;
makoto.morishita.gr@hco.ntt.co.jp

Abstract

In this paper, we describe the submission of Tohoku-AIP-NTT to the WMT'20 news translation task. We participated in this task in two language pairs and four language directions: English↔German and English↔Japanese. Our system consists of techniques such as back-translation and fine-tuning, which are already widely adopted in translation tasks. We attempted to develop new methods for both synthetic data filtering and reranking. However, the methods turned out to be ineffective, and they provided us with no significant improvement over the baseline. We analyze these negative results to provide insights for future studies.

1 Introduction

The joint team of Tohoku University, RIKEN AIP, and NTT (Tohoku-AIP-NTT) participated in the WMT'20 shared news translation task in two language pairs and four language directions: English→German (En→De), German→English (De→En), English→Japanese (En→Ja), and Japanese→English (Ja→En).

At the very beginning of this year's shared task, we planned to employ the following two enhancements at the core of our system. The first enhancement is the noisy synthetic data filtering (Koehn et al., 2018) to better utilize the millions of back-translated synthetic data. However, as we analyze in Section 5.1, this filtering turned out to be ineffective. The second enhancement is the reranking of n-best candidates generated a the model.

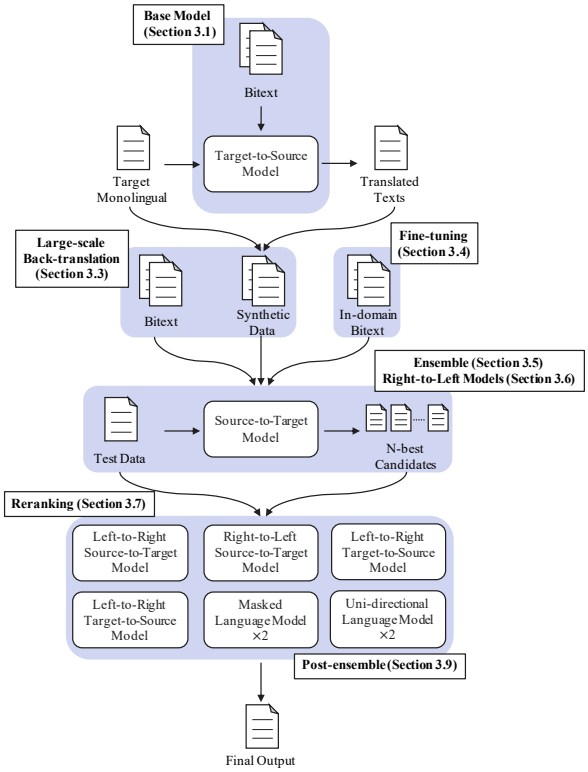


Figure 1: Overview of our system.

Given a collection of scores from multiple generative/translation models, our reranking module selects the best candidate. We attempted to develop sophisticated machine learning based methods for optimizing the weight of each score. However, we found that those methods are not as effective as the simple grid search on the BLEU score (details in Section 3.7 and Section 5.3).

Eventually, we designed our system as a combination of techniques that are already widely adopted in the shared task, such as back-translation and fine-tuning. The overview of our system is shown in Figure 1. We achieved the first place in De→En on automatic evaluation and obtained strong results in other language directions.

[*] Shun conducted most of the experiments for both En↔De and En↔Ja. Takumi preprocessed En↔Ja data. Ryuto trained fasttext word vectors and implemented the post-ensemble method. Takumi and Ryuto worked on synthetic data filtering approaches. Makoto back-translated monolingual corpus for all language directions. Shun, Makoto, and Takumi developed an effective fine-tuning strategy. Jun implemented the entire reranking module and organized the team. Everyone contributed to writing this paper.

Proceedings of the 5th Conference on Machine Translation (WMT), pages 145–155
Online, November 19–20, 2020.

2 Dataset and Preprocessing

2.1 Bitext

For both En↔De and En↔Ja, we used all bitexts that are available for a constrained system.

En↔De Following Ng et al. (2019), we applied language identification filtering (`langid`)[1] to the bitext. In this filtering, sentence pairs were removed if a supposedly English/German sentence is identified as a non-English/German sentence. Then, we applied the `clean-corpus-n` script available in the Moses toolkit (Koehn et al., 2007) and removed sentence pairs that are either too long and/or their length ratio is too large[2]. These two filtering processes provided us with approximately 44M sentence pairs. Then, we trained and applied the Moses `truecaser` independently for each language. We also trained byte-pair encoding (BPE) (Sennrich et al., 2016c) models using the `sentencepiece` (Kudo and Richardson, 2018) implementation. For BPE training, we used only a subset of the parallel corpus (Europarl, NewsCommentary, and RAPID) to prevent extremely rare characters from contaminating the vocabulary and the subword segmentation.

En↔Ja Similar to En↔De, we applied `langid` to clean bitext, but we did not use `clean-corpus-n` since the Japanese text is not segmented. Instead, we simply removed sentence pairs in which the English sentence is longer than 500 tokens. Eventually, we obtained about 17M sentence pairs. We used `truecaser` for the English side only, because case information does not exist in the Japanese language. We independently trained the BPE merge operation on the bitext. We set the character coverage option[3] of `sentencepiece` to 1.0 and 0.9998 for English and Japanese, respectively.

2.2 Monolingual Corpus

The origins of the monolingual corpus in our system are the Europarl, NewsCommentary, and entire NewsCrawl (2008-2019) corpora for English and German, and the Europarl, NewsCommentary and CommonCrawl corpora for Japanese. Similarly to bitext preprocessing in Section 2.1, we applied `langid` filtering to all monolingual corpora. These corpora are used for large-scale back-translation (Section 3.3).

3 System Overview

3.1 Base Model and Hyperparameter

The well-known Transformer model (Vaswani et al., 2017) is our base Encoder Decoder model. Specifically, we started with the "Transformer (big)" setting described by Vaswani et al. (2017) and increased the feed-forward network (FFN) size from 4,096 to 8,192. Ng et al. (2019) reported that this larger FFN setting slightly improves the performance; we also confirmed it in our preliminary experiment.

Table 1 shows a list of hyperparameters for model optimization. We employed an extremely large mini-batch size of 512,000 tokens using the delaying gradient update technique (Bogoychev et al., 2018; Ott et al., 2018). This is because previous studies showed that a large mini-batch size leads to a faster convergence (Ott et al., 2018) and a better generalization (Popel and Bojar, 2018; Bawden et al., 2019; Morishita et al., 2019). We also used a large learning rate of 0.001 to further accelerate the convergence (Goyal et al., 2017; Ott et al., 2018; Liu et al., 2019). We use the `fairseq` toolkit (Ott et al., 2019) for the entire set of experiments. Every reported BLEU score is measured using `SacreBLEU` (Post, 2018).

3.2 Subword Size

For En↔De, we used the subword size of 32,000, which is commonly used in previous studies (Vaswani et al., 2017; Ng et al., 2019). For En↔Ja, we conducted a hyperparameter search for a suitable subword size; Morishita et al. (2019) empirically showed that a small subword size (e.g., 4,000) is superior to those commonly adopted in the literature (e.g., 16,000 and 32,000). Given their findings, we searched for the subword size in the following range: {4000, 8000, 16000, 32000}.

Table 2 shows that the largest subword size achieves the best performance, which is inconsistent with the result of Morishita et al. (2019). One explanation for this result is that Morishita et al. (2019) conducted an experiment on the ASPEC corpus, whose size (approx. 3M) is much smaller than that of the bitext available for the En↔Ja task. That is, the bitext available for the En↔Ja task is sufficiently large for the model to learn a meaningful representation for each subword unit that is

[1] https://github.com/saffsd/langid.py

[2] We set the minimum length to 1, the maximum length to 250, and the maximum ratio to 3.0.

[3] `--character_coverage`

Base Model	
Architecture	Transformer (big) with FFN size of 8,192
Optimizer	Adam ($\beta_1 = 0.9, \beta_2 = 0.98, \epsilon = 1 \times 10^{-8}$)
Learning Rate Schedule	Inverse square root decay
Warmup Steps	4,000
Max Learning Rate	0.001
Dropout	0.3
Gradient Clipping	1.0
Label Smoothing	$\epsilon_{ls} = 0.1$ (Szegedy et al., 2016)
Mini-batch Size	512,000 tokens
Number of Updates	40,000 steps for En↔De and 80,000 steps for En↔Ja
Averaging	Save checkpoint for every 2,000 steps and take an average of last 10 checkpoints
Uni-directional Language Model	
Architecture	`transformer_lm_big` setting available in `fairseq`
Optimizer	Adam ($\beta_1 = 0.9, \beta_2 = 0.98, \epsilon = 1 \times 10^{-8}$)
Learning Rate Schedule	Inverse square root decay
Warmup Steps	4,000
Max Learning Rate	0.0005
Dropout	0.1
Gradient Clipping	1.0
Weight Decay	0.0
Mini-batch Size	512,000 tokens
Number of Updates	50,000 steps
Masked Language Model	
Architecture	RoBERTa-base (Liu et al., 2019)
Optimizer	Adam ($\beta_1 = 0.9, \beta_2 = 0.98, \epsilon = 1 \times 10^{-8}$)
Learning Rate Schedule	Polynomial decay
Warmup Steps	10,000
Max Learning Rate	0.0005
Dropout	0.1
Gradient Clipping	1.0
Weight Decay	0.01
Mini-batch Size	2,048 sentences
Number of Updates	125,000 steps

Table 1: List of hyperparameters for each model.

close to the word level. Thus, we also used the subword size of 32,000 for En↔Ja.

3.3 Large-scale Back-translation

We used the back-translation technique (Sennrich et al., 2016b) to generate large-scale synthetic data. First, we trained models on the bitext for all language pairs. Second, for each language, we fed the monolingual corpus (Section 2.2) to the model. Here, we used the beam search of width 6 and length penalty of 1.0. Finally, we applied length and ratio filtering to the model outputs[4]. The size

[4]For En↔De, we removed sentence pairs that contain sentences longer than 250 tokens. For En↔Ja, we removed sentence pairs such that the English sentence is longer than 250 tokens, or the Japanese sentence is longer than 500 characters.

Subword Size	En→Ja
4,000	19.2
8,000	19.6
16,000	19.4
32,000	19.7

Table 2: Effectiveness of different subword sizes on the validation set of En↔Ja task.

	En→De	De→En	En→Ja	Ja→En
No filtering	336M	236M	1777M	236M
After filtering	328M	230M	235M	230M

Table 3: Number of sentence pairs in the synthetic data of each language pair

of the synthetic data that we generated for each language direction is shown in Table 3. The size of the synthetic data for En→Ja, which is generated from CommonCrawl, is extremely large. Thus, we randomly subsampled the synthetic data of En→Ja so that its size roughly matches those of De→En and Ja→En.

We searched for an effective setting for incorporating the synthetic data. As the most straightforward starting point, we simply combined bitext and synthetic data and trained the model. Here, we upsampled the bitext so that the model sees the bitext and synthetic data at a 1:1 ratio (Ng et al., 2019). Table 4 shows the result. Here, naively using the synthetic data (BASE+BT) decreased the performance of the model trained with the bitext only (BASE). Given this result, we considered the following two enhancements:

Tagged Back-translation We used the tagged back-translation technique (Tagged-BT) (Caswell et al., 2019), which prepends a special tag token (e.g., ⟨BT⟩) to the source sentence of synthetic data. This simple technique can inform the model about the origin of the given training data, i.e., whether the sentence pair is back-translated. Marie et al. (2020) empirically demonstrated that the model trained with such tagged data can avoid overfitting to the synthetic data. In Table 4, the Tagged-BT (BASE+TAGGED-BT) successfully improves the performance from BASE except for the newstest2019. We suspect that the performance does not improve on newstest2019 because it does not contain the "translationese" text, i.e., human-generated translations, which are reported to be the main source of improvement of back-

Setting	newstest 2014	2018	2019
BASE	32.2	47.3	42.2
BASE+BT	32.1	45.9	38.8
BASE+TAGGED-BT	33.0	48.0	42.0
BASE ($l = 9$)+TAGGED-BT	33.1	49.6	42.7
BASE ($l = 12$)+TAGGED-BT	33.4	49.4	42.3

Table 4: Effectiveness of using the synthetic data on En→De

translation (Bogoychev and Sennrich, 2019; Marie et al., 2020).

Deeper Model We also considered increasing the model size to take advantage of a massive amount of training data. Specifically, we increased the number of layers l from 6 to 9 and 12 (Wang et al., 2019). Table 4 shows that the performances of BASE ($l = 9$)+TAGGED-BT and BASE ($l = 12$)+TAGGED-BT are almost comparable. We determined that BASE ($l = 9$)+TAGGED-BT is the best option by considering the model performance and training efficiency regarding the GPU memory constraints.

3.4 Fine-tuning

Fine-tuning the model with an in-domain news corpus is acknowledged as an extremely important technique for boosting the performance (Sennrich et al., 2016b; Junczys-Dowmunt, 2019; Ng et al., 2019; Bawden et al., 2019). We fine-tuned our models as follows:

En↔De For En↔De, we fine-tuned the model with a collection of newstest2008-2018 and evaluated its performance on newstest2019. For En→De, we only used sentence pairs whose source sentence is originally written in English, i.e., we never used texts with translationese on the source side for fine-tuning. Similarly, for De→En, we used sentence pairs whose source sentence is originally written in German. This way, we ensured that our model does not overfit to the translationese texts; since newstest2019 does not contain translationese texts (Barrault et al., 2019), we expected that newstest2020 does not contain translationese either.

We fine-tuned the model for 200 iterations with a mini-batch size of 20,000 tokens. During the fine-tuning, we fixed the learning rate to 1e-06 for De→En and 1e-05 for En→De. We saved the model every 20 iterations and took an average of the last eight saved models for decoding.

En↔Ja For fine-tuning, we used the Kyoto Free Translation Task (KFTT) corpus and NewsCommentary as the *clean* bitext and NewsCommentary as the *news* bitext. We fine-tuned the models by a two-step procedure, that is, we first fine-tuned with the *clean* bitext for 2,000 steps. Then we fine-tuned with the *news* bitext for 200 steps. We found that the validation performance of this two-step procedure is slightly better than that of the fine-tuning with the *news* bitext only.

3.5 Ensemble

We used the model ensemble method to improve the performance. First, we trained four models with different random seeds. These models were then simultaneously used for computing the score of each candidate during the beam search decoding.

3.6 Right-to-Left Models

We used Right-to-Left (R2L) models for reranking the n-best candidates from Left-to-Right (L2R) models. R2L models generate sentences in reverse order. Suppose that conventional L2R models generate sentences from the beginning-of-the-sentence (BOS) to the end-of-the-sentence (EOS); R2L models generate from EOS to BOS. This reranking technique was independently proposed by Liu et al. (2016) and Sennrich et al. (2016a) to mitigate the search error of L2R models, which may occur around EOS. We trained four R2L models and used their scores for reranking the n-best candidates generated by L2R models (Section 3.5). Specifically, we computed the score of each candidate with both L2R models and R2L models. Then, we took the sum of the two scores and obtained the final score. We sorted this final score and then selected the candidate with the highest score.

3.7 Reranking

We also applied a reranking method based on the scores of several translation (or generative) models, which is closely related to one iteration of Minimum Error Rate Training (MERT) (Och, 2003) often used in Statistical Machine Translation (SMT). The underlying idea is to find the balance of likelihood independently computed from the models.

Suppose we have a set of candidate output sentences for each input in either the validation (training phase) or the test (evaluation phase) sets. In our case, we independently generated n-best candidates using the L2R and R2L models, and obtained $2n$ candidates in total for each. Here, let $\mathcal{C}_i$ represent the set of the obtained $2n$ candidates of the

i-th input.

Next, $P_j(e) \in [0, 1]$ denotes the score of the candidate $e \in \mathcal{C}_i$ obtained from the j-th model, where $j \in \{1, \ldots, J\}$. Let $w_j \in [0, 1]$ be a weighting factor of the j-th model, and $\boldsymbol{w} = (w_1, \ldots, w_J)$ be the vector representation of the weighting factor. We then obtained the most likely candidate $\hat{e}_{i,\boldsymbol{w}}$ from $\mathcal{C}_i$ given the i-th input and $\boldsymbol{w}$ as follows:

$$\hat{e}_{i,\boldsymbol{w}} = \underset{e \in \mathcal{C}_i}{\operatorname{argmax}} \left\{ \sum_{j=1}^{J} w_j \log(P_j(e)) \right\}. \quad (1)$$

Finally, for the parameter estimation of $\boldsymbol{w}$, we explored $\widehat{\boldsymbol{w}}$ by using the following optimization problem:

$$\widehat{\boldsymbol{w}} = \underset{\boldsymbol{w} \in \mathcal{G}_{\boldsymbol{w}}}{\operatorname{argmax}} \{ \texttt{SacreBLEU}(\widehat{\mathcal{E}}_{\boldsymbol{w}}) \}, \quad (2)$$

where $\hat{\mathcal{E}}_{\boldsymbol{w}} = (\hat{e}_{i,\boldsymbol{w}})_{i=1}^{I}$ and $\mathcal{G}_{\boldsymbol{w}}$ represent a set of values that w_j can take, namely, $[0, 1]^J$.

For the reranking experiment, we prepared the following generative and translation models to compute $P_j(e)$.

Source-to-Target L2R and R2L Model The Source-to-Target L2R and R2L models are the same as that used for the candidate generation; the ensemble of four L2R models and four R2L models compute the score of each candidate.

Target-to-Source L2R and R2L Model The Target-to-Source (T2S) model translates a sequence in a reverse direction, that is, it translates a given target sequence to a source sequence. For example, if a candidate sentence is generated by the En→De model, we use the De→En model for computing the T2S score.

Uni-directional Language Model We used the uni-directional language model (UniLM) to compute the likelihood of the decoded target sequence. To do this, we trained the Transformer-based language model (Baevski and Auli, 2019) for all languages on monolingual data. We obtained two distinct scores from two normalization methods: (1) simply dividing by the target sequence length (Yee et al., 2019) and (2) *SLOR* (Pauls and Klein, 2012; Lau et al., 2020). A list of hyperparameters is shown in Table 1.

Masked Language Model We also used the pretrained masked language model (MLM) (Devlin et al., 2019) for computing the score. Specifically, we trained the RoBERTa-base (Liu et al., 2019) setting available in `fairseq` on monolingual data. First, we computed the unnormalized log-probabilities by the method described by Wang and Cho (2019). Then, we normalized the probability by (1) dividing by the sequence length and (2) *PenLP* (Vaswani et al., 2017; Lau et al., 2020). A list of hyperparameters is shown in Table 1.

Because the uni-directional language model and MLM both have two distinct variations, we used a total of six models, namely, $J = 6$.

3.8 Post-processing

We converted the decoded target sequence from a sequence of subwords to tokens. Then we applied the Moses `detruecaser` to English and German sequences. We also applied language-specific post-processing as follows:

En↔De We observed that the rare tokens such as Greek letters in the source sequence are sometimes translated into ⟨UNK⟩. We handled ⟨UNK⟩ in the decoded sequence by copying the corresponding token from the source sequence. We determined the corresponding token by finding the token that does not exist in one of the source-side or target-side vocabularies.

En→Ja We did not take any special measures for ⟨UNK⟩[5]. We replaced the English style comma ", " and period ". " with the Japanese style "、" and "。" respectively.

Ja→En We observed that the model translates the Japanese vertical bar " ｜ " to ⟨UNK⟩. Thus, we replaced all ⟨UNK⟩ with "|".

3.9 Post-ensemble

Kobayashi (2018) proposed the method of taking the ensemble of multiple models *after* decoding the sequence, namely, post-ensemble (POSTENSEMBLE). The underlying idea of POSTENSEMBLE is to choose "majority-like" candidates by comparing the similarities among candidates. He applied POSTENSEMBLE to the abstractive summarization task and reported that the performance is superior to that of the conventional ensemble.

We used POSTENSEMBLE in En→Ja[6]. Specifically, we adopted the `PostCosE` variant in which the cosine similarity is used as a similarity metric. We created 300 dim fasttext word vectors (Bojanowski et al., 2017) on the Japanese monolingual corpus.

[5] In fact, we never observed ⟨UNK⟩ in the decoded test set.

[6] Kobayashi (2018) introduced POSTENSEMBLE as the method that *replaces* the conventional ensemble. Instead, we used two ensemble methods simultaneously.

4 Results

Performance on the Validation Set We show the validation performance of our system in Table 5. We used newstest2019 and the official validation set for En↔De and En↔Ja, respectively, for the validation data. The table shows the effectiveness of incorporating each technique described in Section 3. Each technique consistently improves the performance in most cases. In addition, it is noteworthy that both En→De and De→En models significantly outperform the performance of the best system from last year's shared task (WMT'19).

Performance on the Test Set We show the test set performance that we measured in the OCELoT system[7] in Table 6. The system provides us with the SacreBLEU score and the chrF score (Popović, 2015).

We used the following models for POSTENSEMBLE of Ja→En: (1) model (f) (Table 5), (2) Model (f) with the ensemble of eight models, in which four models are fine-tuned with the *clean* bitext and the other four models are fine-tuned with the *news* bitext, and (3) Model (2) without n-best candidates from the R2L model.

The performance of En→Ja appears significantly better than the validation performance reported in Table 5; this is because OCELoT computes the BLEU score with character-level segmentation, whereas we used the MeCab-based word-level segmentation[8]. We also computed the BLEU score with the MeCab-based segmentation for reference and obtained 25.8 points.

5 Analysis

In this section, we introduce several negative results from our preliminary experiments. Our attempts include the following: (1) filtering synthetic data, (2) incorporating forward-translation, and (3) developing a more sophisticated reranking method. We also analyze the issue regarding the use of brackets in the En→Ja task.

5.1 Negative Results on Synthetic Data Filtering

We applied corpus filtering to the synthetic data created in Section 3.3. The goal of this filtering is to extract and utilize the "clean" subset of synthetic data that may contribute to the model performance. For each of the sentence pairs in the synthetic data, we assigned scores that represent the likelihood of being a sentence pair (Section 5.1.1). Then, we regarded these scores as features for classification; we trained a model classifying clean and noisy sentence pairs (Section 5.1.2). Finally, on the basis of the confidence scores of the classifier, we extracted the presumably clean subset of the synthetic data.

5.1.1 Features

Pointwise HSIC We computed the score for each sentence pair using the pointwise Hilbert-Schmidt independence criterion (PHSIC) (Yokoi et al., 2018), which is a kernel-based co-occurrence measure. Given a set of sentence pairs, PHSIC can assign a high score to a sentence pair that is consistent with the rest of the sentence pairs. To do this, PHSIC utilizes kernel functions and calculates the sentence similarity. Yokoi et al. (2018) applied PHSIC to machine translation corpus filtering and reported promising results. Thus, we also employed PHSIC for synthetic data filtering.

First, we learned the parameters of the PHSIC matrix with a cosine kernel by using all sentence pairs in the bitext, which are represented as sentence embeddings. Then, we used this trained matrix to compute the scores for the synthetic data. We used the following two methods for computing the sentence embeddings: (1) the weighted sum of fasttext vectors (Bojanowski et al., 2017) by smoothed inverse frequency (SIF) weighting (Arora et al., 2017) and (2) the average of final hidden states of the pre-trained MLM. Here, the fasttext vector is the same as the one used for post-ensemble (Section 3.9), and MLM is the one from the reranking (Section 3.7). The word frequency for SIF weighting is calculated from the monolingual corpus.

Cross-entropy from T2S Model We computed the word-normalized conditional cross-entropy using the T2S translation model. For example, the synthetic data generated using the En→De model are scored using the De→En model.

5.1.2 Training a Classifier

We trained a linear support vector machine model that classifies clean and noisy sentence pairs. To train the classifier, we used newstest2009-2019 and the official validation set as clean sentence pairs for En↔De and En↔Ja, respectively. We generated the noisy sentence pairs by randomly adding the noise presented by Wang et al. (2018) to the clean sentence pairs.

[7] https://ocelot.mteval.org/

[8] The use of the MeCab-based segmentation is recommended by SacreBLEU.

ID	Setting	En→De	De→En	En→Ja	Ja→En
(a)	BASE (Section 3.1)	42.4	42.0	19.7	21.6
(b)	BASE ($l = 9$)+TAGGED-BT (Section 3.3)	42.7	42.5	22.0	23.9
(c)	(b) + fine-tuning (Section 3.4)	44.9	42.3	23.1	24.4
(d)	(c) × 4 (Section 3.5)	45.5	42.8	23.9	25.4
(e)	(d) + 4 × (c)-R2L (Section 3.6)	45.4	43.6	24.2	25.9
(f)	(e) + reranking (Section 3.7)	45.7	43.8	24.9	26.2
-	The best system in WMT'19	44.9	42.8	-	-

Table 5: Effectiveness of each technique: we use newstest2019 and official validation set for En↔De and En↔Ja respectively. The best result from WMT'19 is unavailable for En↔Ja, because this task has newly appeared this year.

Direction	Setting / ID	BLEU	chrF
En→De	(f) (Table 5)	37.5	0.647
De→En	(f) (Table 5)	43.8	0.690
En→Ja	(f) (Table 5)	40.1	0.343
Ja→En	POSTENSEMBLE	25.5	0.536

Table 6: Performance on WMT'20 Test Set: refer to Table 5 for model ID.

After training, we classified each sentence pair in the synthetic data. The confidence score of the classifier was used as an overall score that represents the "cleanness" (i.e., quality) of the sentence pair.

5.1.3 Results

We investigated the effectiveness of the synthetic data filtering. First, we sorted the synthetic data according to the score computed with the classifier (Section 5.1.2). Then, we used the top $r\%$ of synthetic data for training.

Table 7 shows the results of synthetic data filtering with varying r. We trained the En→De model using the BASE+TAGGED-BT setting. The results showed that our filtering does not seem to improve the performance over the baseline ($r = 100$). One of the possible reasons for this ineffectiveness is the quality of the sentence embeddings used for PHSIC. That is, the use of fasttext and pretrained MLM might be inappropriate. Utilizing more powerful sentence encoders such as Sentence-BERT (Reimers and Gurevych, 2019) and Universal Sentence Encoder (Cer et al., 2018) is an interesting option to explore in the future; however, the methods of acquiring such resources in the constrained setting is not trivial.

	newstest		
Amount of Synthetic Data Used: r (%)	2014	2018	2019
100	33.0	48.0	42.0
50	32.9	48.4	42.3
33	33.1	47.9	42.2
25	32.9	48.5	42.4

Table 7: Effectiveness of corpus filtering on En→De.

	newstest		
Setting	2014	2018	2019
BASE	32.2	47.3	42.2
BASE+TAGGED-BT	33.0	48.0	42.0
BASE+TAGGED-FT	31.7	46.7	42.1
BASE+TAGGED-BT+TAGGED-FT	33.1	48.3	42.4

Table 8: Effectiveness of incorporating forward-translation and back-translation on En→De.

5.2 Effectiveness of Incorporating Forward-Translation

Forward-translation (Burlot and Yvon, 2018) is a technique similar to back-translation; the difference is that while back-translation uses the target-side monolingual data, forward-translation uses the source-side monolingual data to generate synthetic data. Bogoychev and Sennrich (2019) reported that forward-translation is effective for improving the translation of texts that are originally written in the source language (i.e., non-translationese texts).

To determine if we can take the best of the two techniques, namely, forward-translation and back-translation, we combined the synthetic data and trained the model. As described in Section 3.3, we prepended a distinct tag to each data source: ⟨FT⟩ and ⟨BT⟩ for data generated by forward-translation and back-translation respectively. Then, we upsampled the bitext, so that the model is fed with the bitext and synthetic data at a 1:0.5:0.5 ratio.

Table 8 shows the result. The model in-

Input	Only one member of the family, then 15-year-old Cassidy Stay, survived.
Reference	家族の中で、ただ一人、当時15歳だったカシディ・ステイさんだけが一命を取り留めた。
Model Output	当時15歳のキャシディ・ステイ(Cassidy Stay)だけが生き残った。
Input	Madam Needjan, pledged the association's support to the hospital and called on other associations to emulate the gesture.
Reference	マダム・ニージャンは、協会の当病院への支援を約束し、他の団体もこうした行為に追随するよう呼びかけた。
Model Output	マダム・ニージャン(Madam Needjan)は、協会が病院を支援することを約束し、他の協会にこのジェスチャーを模倣するよう求めた。

Figure 2: Error analysis of En→Ja translation.

corporating both back-translation and forward-translation (BASE+TAGGED-BT+TAGGED-FT) achieves the best result, however, the improvement was marginal. In addition, the performance of the model with forward-translation only (BASE+TAGGED-FT) was worse than that of the baseline (BASE) in all datasets. Given this result, we only used back-translation and kept the training procedure as simple as possible in our final system.

5.3 Negative Result on Reranking

We actually investigated several different types of reranking algorithms other than the standard grid search described in Section 3.7. For example, we experiment withed optimizing model weights by machine learning based methods such as those using support vector machines, XGBoost (Chen and Guestrin, 2016), and deep neural networks. Unfortunately, none of them worked well. In this competition, we only used the model scores for the reranking. This setting immediately leads the overfitting to the development sets, and hard to extract meaningful generalized weights (rules) that also work well for unseen test data. The development of the methods that can further and consistently improve the quality of translations is our future work for the next year.

5.4 Japanese Text and Brackets

Figure 2 shows examples from the validation set of the En→Ja task. These examples illustrate the weakness of our model, in which the named entities are often inappropriately translated. According to the references in the figure, the named entities must be translated from alphabetical characters to *katakana* (カタカナ), e.g., *Cassidy Stay* to カシディ・ステイ. Although our model successfully translates the named entities in most of the cases, the model also copies original alphabetical characters into the brackets. For example, the model translates *Madam Needjan* to マダム・ニージャン*(Madam Needjan)*. These alphabetical characters damage the BLEU score. We can remove the extra brackets by the rule-based post-processing; however, we find that this naive operation hurts the brevity penalty.

This extra bracket problem seems to reflect the way that the named entities are written in the En↔Ja training data such as KFTT. We should have considered special preprocessing measures in advance to alleviate this problem.

6 Conclusion

In this paper, we described the submission of the joint team of Tohoku, AIP, and NTT (Tohoku-AIP-NTT) to the WMT'20 news translation task. We participated in the En↔De and En↔Ja translation. In preliminary experiments, we attempted new techniques such as synthetic data filtering, forward-translation, and sophisticated reranking. However, none of them was effective. In the submission, we used several standard techniques such as back-translation and fine-tuning. As a result, we achieved the best BLEU score on De→En and strong results in other directions.

Acknowledgments

We would like to thank the two anonymous reviewers for their valuable feedback.

References

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A Simple but Tough-to-Beat Baseline for Sentence Embeddings. In *Proceedings of the 5th International Conference on Learning Representations (ICLR 2017)*.

Alexei Baevski and Michael Auli. 2019. Adaptive Input Representations for Neural Language Modeling. In *Proceedings of the 7th International Conference on Learning Representations (ICLR 2019)*.

Loïc Barrault, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. 2019. Findings of the 2019 Conference on Machine Translation (WMT19). In *Proceedings of the Fourth Conference on Machine Translation (WMT 2019)*, pages 1–61.

Rachel Bawden, Nikolay Bogoychev, Ulrich Germann, Roman Grundkiewicz, Faheem Kirefu, Antonio Valerio Miceli Barone, and Alexandra Birch. 2019. The University of Edinburgh's Submissions to the WMT19 News Translation Task. In *Proceedings of the Fourth Conference on Machine Translation (WMT 2019)*, pages 103–115.

Nikolay Bogoychev, Kenneth Heafield, Alham Fikri Aji, and Marcin Junczys-Dowmunt. 2018. Accelerating Asynchronous Stochastic Gradient Descent for Neural Machine Translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP 2018)*, pages 2991–2996.

Nikolay Bogoychev and Rico Sennrich. 2019. Domain, Translationese and Noise in Synthetic Data for Neural Machine Translation. *arXiv preprint arXiv:1911.03362*.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics (TACL 2017)*, 5:135–146.

Franck Burlot and François Yvon. 2018. Using Monolingual Data in Neural Machine Translation: a Systematic Study. In *Proceedings of the Third Conference on Machine Translation (WMT 2018)*, pages 144–155.

Isaac Caswell, Ciprian Chelba, and David Grangier. 2019. Tagged Back-Translation. In *Proceedings of the Fourth Conference on Machine Translation (WMT 2019)*, pages 53–63.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. 2018. Universal Sentence Encoder for English. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 169–174.

Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 785–794. ACM.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2019)*, pages 4171–4186.

Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. 2017. Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour. *arXiv preprint arXiv:1706.02677*.

Marcin Junczys-Dowmunt. 2019. Microsoft Translator at WMT 2019: Towards Large-Scale Document-Level Neural Machine Translation. In *Proceedings of the Fourth Conference on Machine Translation (WMT 2019)*, pages 225–233.

Hayato Kobayashi. 2018. Frustratingly Easy Model Ensemble for Abstractive Summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP 2018)*, pages 4165–4176.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180.

Philipp Koehn, Huda Khayrallah, Kenneth Heafield, and Mikel L. Forcada. 2018. Findings of the WMT 2018 Shared Task on Parallel Corpus Filtering. In *Proceedings of the Third Conference on Machine Translation (WMT 2018)*, pages 726–739.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71.

Jey Han Lau, Carlos Armendariz, Matthew Purver, Chang Shu, and Shalom Lappin. 2020. How Furiously Can Colourless Green Ideas Sleep? Sentence Acceptability in Context. *Transactions of the Association for Computational Linguistics*, 8:296–310.

Lemao Liu, Masao Utiyama, Andrew Finch, and Eiichiro Sumita. 2016. Agreement on Target-bidirectional Neural Machine Translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2016)*, pages 411–416.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv preprint arXiv:1907.11692*.

Benjamin Marie, Raphael Rubino, and Atsushi Fujita. 2020. Tagged Back-translation Revisited: Why Does It Really Work? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020)*, pages 5990–5997.

Makoto Morishita, Jun Suzuki, and Masaaki Nagata. 2019. NTT Neural Machine Translation Systems at WAT 2019. In *Proceedings of the 6th Workshop on Asian Translation (WAT 2019)*, pages 99–105.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook FAIR's WMT19 News Translation Task Submission. In *Proceedings of the Fourth Conference on Machine Translation (WMT 2019)*, pages 314–319.

Franz Josef Och. 2003. Minimum Error Rate Training in Statistical Machine Translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL 2003)*, pages 160–167.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A Fast, Extensible Toolkit for Sequence Modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53.

Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. 2018. Scaling Neural Machine Translation. In *Proceedings of the Third Conference on Machine Translation (WMT 2018)*, pages 1–9.

Adam Pauls and Dan Klein. 2012. Large-Scale Syntactic Language Modeling with Treelets. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL 2012)*, pages 959–968.

Martin Popel and Ondřej Bojar. 2018. Training Tips for the Transformer Model. *The Prague Bulletin of Mathematical Linguistics*, 110(1):43–70.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395.

Matt Post. 2018. A Call for Clarity in Reporting BLEU Scores. In *Proceedings of the Third Conference on Machine Translation (WMT 2018)*, pages 186–191.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP 2019)*, pages 3982–3992.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Edinburgh Neural Machine Translation Systems for WMT 16. In *Proceedings of the First Conference on Machine Translation (WMT 2016)*, pages 371–376.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Improving Neural Machine Translation Models with Monolingual Data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*, pages 86–96.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016c. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*, pages 1715–1725.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the Inception Architecture for Computer Vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016)*, pages 2818–2826.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. In *Advances in Neural Information Processing Systems 31 (NIPS 2017)*, pages 5998–6008.

Alex Wang and Kyunghyun Cho. 2019. BERT has a Mouth, and It Must Speak: BERT as a Markov Random Field Language Model. In *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation*, pages 30–36.

Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F. Wong, and Lidia S. Chao. 2019. Learning Deep Transformer Models for Machine Translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL 2019)*, pages 1810–1822.

Rui Wang, Benjamin Marie, Masao Utiyama, and Eiichiro Sumita. 2018. NICT's Corpus Filtering Systems for the WMT18 Parallel Corpus Filtering Task. In *Proceedings of the Third Conference on Machine Translation (WMT 2018)*, pages 963–967.

Kyra Yee, Yann Dauphin, and Michael Auli. 2019. Simple and Effective Noisy Channel Modeling for Neural Machine Translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP 2019)*, pages 5696–5701.

Sho Yokoi, Sosuke Kobayashi, Kenji Fukumizu, Jun Suzuki, and Kentaro Inui. 2018. Pointwise HSIC: A Linear-Time Kernelized Co-occurrence Norm for Sparse Linguistic Expressions. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP 2018)*, pages 1763–1775.

NRC Systems for the 2020 Inuktitut–English News Translation Task

Rebecca Knowles, Darlene Stewart, Samuel Larkin, and **Patrick Littell**
National Research Council Canada
{Rebecca.Knowles, Darlene.Stewart, Samuel.Larkin, Patrick.Littell}@nrc-cnrc.gc.ca

Abstract

We describe the National Research Council of Canada (NRC) submissions for the 2020 Inuktitut–English shared task on news translation at the Fifth Conference on Machine Translation (WMT20). Our submissions consist of ensembled domain-specific finetuned transformer models, trained using the Nunavut Hansard and news data and, in the case of Inuktitut–English, backtranslated news and parliamentary data. In this work we explore challenges related to the relatively small amount of parallel data, morphological complexity, and domain shifts.

1 Introduction

We present the National Research Council of Canada (NRC) Inuktitut–English[1] machine translation (MT) systems in both translation directions for the 2020 WMT shared task on news translation.

Inuktitut is part of the dialect continuum of Inuit languages, the languages spoken by Inuit, an Indigenous people whose homeland stretches across the Arctic. Included in this continuum are Indigenous languages spoken in northern Canada, including but not limited to the Territory of Nunavut. The term *Inuktut* is used by the Government of Nunavut (2020) to describe Inuit languages spoken in Nunavut, such as Inuktitut and Inuinnaqtun. The majority of the Inuit language text provided for the shared task comes from ᓄᓇᕗᑦ ᒪᓕᒐᓕᐅᕕ (Nunavut Maligaliurvia; Legislative Assembly of Nunavut) through the Nunavut Hansard, the published proceedings of the Legislative Assembly of Nunavut. The Nunavut Hansard is released publicly by the Government of Nunavut in Inuktitut and English (also an official language of Nunavut), and with their generous assistance was recently processed and released for use in building MT systems (Joanis et al., 2020).[2]

In this work, we examined topics related to morphological complexity and writing systems, data size, and domain shifts. Our submitted systems are ensembled domain-specific finetuned transformer models, trained using Nunavut Hansard and news data and, in the case of Inuktitut–English, backtranslated news and parliamentary data. We measured translation performance with BLEU (Papineni et al., 2002),[3] metrics specific to the production of Roman text in Inuktitut, and human evaluation (to be reported). We hope that human evaluation will provide insight as to whether the current state of the art is sufficient to start building computer aided translation tools of interest and use to Inuit language translators, or whether more work is needed to make the systems usable.

2 Related Work and Motivation

Initial experiments on building neural machine translation (NMT) systems for Inuktitut–English using the most recent Nunavut Hansard corpus are reported in Joanis et al. (2020). Earlier work includes Micher (2018) and Schwartz et al. (2020), and, predating the recent wave of NMT, Martin et al. (2003), Martin et al. (2005), and Langlais et al. (2005). There has also been work on morphological analysis of Inuktitut, including Farley

[1] Abbreviated `iu` and `en` using ISO 639-2 codes.

[2] Though we note that Inuktitut, Inuinnaqtun, English, and French may all be spoken in the House, we use the term Inuktitut in describing our MT systems for two main reasons: 1) the official website describes the Nunavut Hansard as being published "in both Inuktitut and English" (Legislative Assembly of Nunavut, 2020) and 2) because we wish to make clear the limitations of our work; there is no reason to expect that the systems built using the data provided for WMT will perform well across various Inuit languages and dialects (or even across a wider range of domains).

[3] Computed using `sacreBLEU` version 1.3.6 (Post, 2018) with mteval-v13a tokenization: `BLEU+case.mixed+numrefs.1+smooth.exp+tok.13a`.

Proceedings of the 5th Conference on Machine Translation (WMT), pages 156–170

(2009) and Micher (2017). In this work, we focus mainly on approaches that are not language-specific, but that are motivated by specific challenges of translating relatively low-resource, morphologically complex languages; thus they are also not entirely language-agnostic.

2.1 Language Typology and Writing Systems

Inuit languages are highly morphologically-complex; many Inuktitut words consist of a large number of morphemes, and can translate to entire phrases or clauses in English (Mallon, 2000; Micher, 2017; Joanis et al., 2020).[4]

Moreover, these morphemes are not easily segmented from one another, as they exhibit phonological changes at morpheme boundaries. That is to say, a given morpheme may be spelled in a number of different ways (or may even appear to merge with a neighbouring morpheme) depending on the morphemes adjacent to it. This means that automatic segmentation approaches may not be optimal. Nevertheless we try using them, and see if we can mitigate some of those challenges via experiments on joint vs. disjoint vocabularies and inserting noise into the segmentation process.

English is written in the Roman script (ISO 15924: Latn), while the Inuktitut data used for this task is primarily written in syllabics (ISO 15924: Cans).[5] There is some Roman text in the Inuktitut side of the data and some syllabics text in the English side of the data, though the former is much more common than the latter.

2.2 Domains and Recency Effects

The Inuktitut–English training corpus released for WMT 2020 consists of parliamentary transcriptions and translations from the Legislative Assembly of Nunavut (Joanis et al., 2020), while the development and test sets are a mix of parliamentary text and news text, the latter drawn from Nunatsiaq News.[6] These two domains are quite different from one another, and in our initial baseline experiments (training only on parliamentary data), we observed very low BLEU scores when translating news data. As we wished to build a constrained system, our only source of Inuktitut news was the data in the development set. In order to retain the ability to use news data in the development and test sets, we utilized an approach of dividing the news development data into thirds, including a third in the training set, using a third as part of the validation set, and holding the remaining third out as test.

The Nunavut Hansard is known to exhibit recency effects, i.e., when testing on a recent subset of the corpus, training on a recent subset is better than training on an early subset (Joanis et al., 2020). Although we have not fully examined the reasons behind this, it could be due to topic shift, a shift in the named entities in the corpus, changes in transcription and translation practices, or any combination of these and more.

We consider tagging domain as one approach to this. Sennrich et al. (2016a) use side constraints (tags) in order to control English–German MT output formality. Yamagishi et al. (2016) add information about voice (e.g. active/passive) to Japanese–English translation via source-side tags. Johnson et al. (2017) also use tags at the start of source sentences, in their case to indicate what language the multilingual translation system should translate into.[7] One might consider domain to fall somewhere between these use cases; Kobus et al. (2017) use domain tags to influence translation in a multi-domain setting. Caswell et al. (2019) use tags to indicate when data has been backtranslated.

3 Data

While the parallel text size for this language pair is quite small compared to high-resource language pairs in the news translation task, Inuktitut is one of the few Indigenous languages in Canada (or possibly the only) for which there exists enough parallel text (with any other language) to train robust statistical or NMT systems outside of the strictly low-resource paradigm (Littell et al., 2018). Thus we expect that it should be helpful to incorporate available monolingual data.

We trained our baseline models using the full 1.3 million line Nunavut Hansard 3.0 (NH) parallel corpus. For IU-EN, we also used a random subselection of 1.3M sentences of English Europarl v10 (Koehn, 2005; Tiedemann, 2012) and 1.3M sentences of English 2019 News data[8] backtranslated into Inuktitut (Section 5.4). We did not use

[4] The Inuktut Tusaalanga website provides an overview of grammar: https://tusaalanga.ca/node/1099

[5] A number of different writing systems, including both syllabics and Roman orthography, are used to write Inuit languages. Inuit Tapiriit Kanatami (ITK) is in the process of creating a unified writing system (Inuit Tapiriit Kanatami, 2020).

[6] https://nunatsiaq.com/

[7] Wang et al. (2018) add a target-side tag.

[8] From the WMT 2020 task page: https://www.statmt.org/wmt20/translation-task.html

Wiki Titles or Common Crawl Inuktitut data.[9] We incorporated the news portion of the development data in training our models to alleviate the domain mismatch issue (Section 5.1).

4 Preprocessing and Postprocessing

We first applied an internal script to convert control characters to spaces as well as normalizing spaces and hyphens; this was effectively a no-op for the Nunavut Hansard parallel corpus, but removed some problematic characters in the monolingual training data. Parallel training corpora were then cleaned with the Moses `clean-corpus-n.perl` script (Koehn et al., 2007), using a sentence length ratio of 15:1 and minimum and maximum lengths of 1 and 200, respectively. For monolingual training data, the second cleaning step consisted of removing empty lines. For Inuktitut, we used the `normalize-iu-spelling.pl` script provided by the organizers.

We then performed punctuation normalization. This included specific `en` and `iu` normalization scripts, to more accurately capture and retain information about directional quotation marks, different types of dashes, and apostrophes, normalizing to the most common form. For Inuktitut, this included treating word-internal apostrophes as U+02BC MODIFIER LETTER APOSTROPHE.[10] Appendix C provides a detailed description. After this preliminary normalization, we applied the Moses `normalize-punctuation.perl` script, with the language set to `en` (or backing off to `en`, as there are currently no Inuktitut-specific rules implemented).

Having noted that some of the lines in the training data contained more than one sentence (which results in unintended tokenization behavior), we next performed sentence splitting using the Portage sentence splitter (Larkin et al., 2010) on each side of the training data before feeding it to the Moses tokenizer (using aggressive hyphen splitting). Sentences that had been split were then re-merged following tokenization.

We trained joint byte-pair encoding (BPE; Sennrich et al., 2016c) models on the full Nunavut Hansard parallel training data using `subword-nmt`, then extracted English and Inuktitut vocabularies separately.[11] Using a joint BPE model improves performance on Roman text in Inuktitut output (Section 5.2 and Appendix B).

As postprocessing, we de-BPE the data, run the Moses detokenizer, and then convert the placeholder tokens from our normalization scripts to their corresponding symbols (dashes, apostrophes, quotation marks, etc.).[12]

5 Experiments

All models were typical transformers (Vaswani et al., 2017) with 6 layers, 8 attention heads, network size of 512 units, and feedforward size of 2048 units, built using Sockeye (Hieber et al., 2018) version 1.18.115. We have changed the default gradient clipping type to absolute, used the whole validation set during validation, an initial learning rate of 0.0001, batches of ∼8192 tokens, and maximum sentence length of 200 tokens. We have optimized for BLEU. Custom checkpoint intervals have been used during training, with final systems using between 2 and 11 checkpoints per epoch, consistent within sets of experiments (e.g., vocabulary size sweeping). For finetuning, the checkpoint interval is set to 9, resulting in about 2 checkpoints per epoch for news and 13 for Hansard. For finetuning, we used an initial learning rate of 0.00015 (decreasing by a factor of 0.7 if there was no improvement after 8 checkpoints). Decoding was done with beam size 5.

In the following sections, we describe the experiments that led to our submitted systems. Our final systems were trained on a mix of news and Hansard data (Section 5.1), using joint BPE (Section 5.2), BPE-dropout (for EN-IU; Section 5.3), tagged backtranslation (for IU-EN; Section 5.4), finetuning (Section 5.5), ensembling, and the use of domain-specific models (Section 5.6).

[9] Appendix E provides additional detail about noise and other concerns with the Common Crawl data.

[10] The apostrophe sometimes represents a glottal stop, so when it appeared between syllabic characters, we treated it as a letter that should not be tokenized.

[11] When extracting the BPE vocabulary (which we then used consistently for all experiments) and when applying the BPE model, we used a glossary containing the special tokens produced in preprocessing, Moses special tokens, and special tags (Section 5.4), to ensure they would not be split.

[12] During the task test period, we noted that the test data contained spurious quotation marks, wrapping some entire sentences. After notifying the organizers and confirming that those were produced in error, we handled them as followed: removed the straight quotes that surrounded complete lines, preprocessed, translated, and postprocessed the text that had been contained inside of them, and then reapplied the quotes to the output. There is not an exact match between the source and target for these spurious quotes, so this approach is effective but *not* an oracle.

5.1 Training and Development Splits

In baseline experiments, training only on the Nunavut Hansard training data provided, we noted a major difference in BLEU scores between the Hansard and news portions of the development set. While BLEU scores should not be compared directly across different test sets, the magnitude of this difference (in the EN-IU direction, BLEU scores in the mid-20s on Hansard and in the mid-single digits on news) and the knowledge of differences between parliamentary speech and the news domain suggested that there was a real disparity, likely driven by train/test domain mismatch.

To test this we divided the news portion of the development set in half, maintaining the first half as development data, and adding the second half to the training corpus. Adding half the news nearly doubled the BLEU score on the held out half of the news data, if we duplicated it between 5 and 50 times (to account for how much more Hansard data was available).[13] Initial experiments on vocabulary types and sizes were performed in this setting (Section 5.2).

For the remainder of our experiments, we switched to a setting where we divided the news data into three approximately equally sized thirds; to maintain most documents separate across splits, we split the data into consecutive chunks. Most experiments were run with the first third added to training data, the second third as part of the development set alongside the Hansard development set, and the final third as a held-out test set. This permitted additional experiments on finetuning (Section 5.5) with a genuinely held-out test set.[14] For our final systems, we ensembled systems that had been trained on each of the thirds of the news development data.

5.2 BPE

Ding et al. (2019) highlight the importance of sweeping the number of subword merges (effectively, vocabulary size) parameter, particularly in lower-resource settings. We swept a range of disjoint BPE size pairs (see Appendix A for details of vocabulary size and sweep), and found that disjoint 1k vocabularies performed well for IU-EN, while the combination of disjoint 5k (EN) and 1k (IU) vocabularies performed well for EN-IU (on the basis of averaged Hansard development and news development BLEU score).

As noted in Section 2.1, the Inuktitut data is written in syllabics. However, it contains some text in Roman script, in particular, organization names and other proper nouns. Over 93% of the Roman tokens that appear in the Inuktitut development data also appear in the corresponding English sentence. The ideal behavior would be for a system to copy such text from source to target. When the BPE vocabulary model is learned jointly the system can learn a mapping between identical source and target tokens, and then learn to copy. When the vocabulary is disjoint, there may not be identical segmentations for the system to copy, posing more of a challenge. Appendix B provides details of our experiments on joint vocabulary for successfully producing Roman text in Inuktitut output.

Due to the similarity in BLEU scores, and for simplicity and consistency, the remainder of our experiments *in both directions* were performed with jointly learned (and separately extracted) BPE vocabularies. We experimented with joint BPE vocabulary sizes of 1k, 2k, 5k, 10k and 15k.

5.3 BPE-Dropout

Knowing that the morphology of Inuktitut may make BPE suboptimal, we chose to apply BPE-dropout (Provilkov et al., 2020) as implemented in `subword-nmt` in an attempt to improve performance. BPE-dropout takes an existing BPE model, and when determining the segmentation of each token in the data randomly drops some merges at each merge step. The result is that the same word may appear in the data with multiple different segmentations, hopefully resulting in more robust subword representations. Rather than modifying the NMT system itself to reapply BPE-dropout during training, we treated BPE-dropout as a preprocessing step. We ran BPE-dropout with a rate of 0.1 over both the source and target training data 5 times using the same BPE merge operations, vocabularies and glossaries as before, concatenating these to form a new extended training set.[15]

In our initial baseline experiments (without

[13] Adding all of the data would not have allowed us to evaluate the outcome on news data, and not including any news data in the development set also hurt performance.

[14] An alternative approach would be to select pseudo in-domain data from the Hansard, by finding the Hansard data that is most similar to the news data (Axelrod et al., 2011; van der Wees et al., 2017). While this may be worth exploring, we felt the extreme discrepencies between news and Hansard merited examination with gold in-domain data.

[15] We also experimented with 11 and 21 duplicates of the training data, and dropout rates of 0.2; we did not observe major differences between the settings.

news data in training), we found that BPE-dropout was more helpful in the IU-EN direction (+∼0.4 BLEU) than in the reverse (+∼0.2 BLEU). After incorporating a third of the news data in training, we found the reverse: a small increase for IU-EN (+∼0.1) and a slightly larger increase for EN-IU (+∼0.3).

5.4 Tagging and Backtranslation

By incorporating news data into our training set (Section 5.1), we improve performance on news data. However, the approach is sensitive to the number of copies of news data added, which can decrease performance on both Hansard and news data if not carefully managed. Both English news data and monolingual English parliamentary data (from Europarl) are plentiful in WMT datasets, so we incorporated them into our models via backtranslation (Sennrich et al., 2016b).

We apply approaches from Kobus et al. (2017) and Caswell et al. (2019): tagging source data domain (<NH> or <NEWS>) and (for IU-EN) tagging backtranslated source data (<BT>). Tagging domain appears to be particularly important for translating into Inuktitut, with between 1.4 and 2.4 BLEU points improvement on a subset of the news development test and minimal effect on the Hansard development data scores.

For backtranslation, we chose random samples of Europarl and WMT 2019 news data, experimenting with 325k, 650k, and 1.3M lines each, with 1.3 million performing best.[16] Ablation experiments with just news or just Europarl data showed less promise than the two combined. We did not perform backtranslation of Inuktitut (see Appendix E).

We performed two rounds of backtranslating the randomly sampled 1.3M lines each of Europarl and WMT 2019 news data. The first round (BT1) used our strongest single 5k joint BPE (with dropout) EN-IU system at the time. The second round (BT2) used a stronger three-way ensemble, with improved performance on both Hansard and news.

We experimented with combinations of tags for the backtranslated data (other parallel corpora have source domain tags unless otherwise stated):

- tagging all backtranslated data with <BT>;
- tagging backtranslated data with both <BT> and domain tags, where the domain tag matches the closest parallel corpus domain, i.e., <BT> <NH> or <BT> <NEWS>.[17]
- tagging backtranslated data with just a domain tag matching the closest parallel corpus domain, i.e. <NH> or <NEWS>.
- tagging all backtranslated data with <BT>, but not domain tagging the parallel corpora.
- tagging nothing.

As Table 1 shows for IU-EN translation, using backtranslated Europarl and news text clearly helped translating news text (as much as 8.0 BLEU) while only slightly impacting the translation of Hansard text. Without any backtranslated text, using domain tags (<NH> for Hansard and <NEWS>) appears to have a small positive effect on Hansard translation, and none on news (contrary to what we observed in the EN-IU direction).

The main observation from these experiments was that it was most important to distinguish backtranslation from true bitext (an observation similar to those noted in Marie et al. (2020)). Our best results were observed with no tags for the bitext and the <BT> tag for the backtranslated data. These experiments finished after the deadline, so our final submission uses the the next best combination: domain tags for bitext and <BT> tags for backtranslation.[18]

5.5 Finetuning

After building models with domain tags and backtranslation (in the case of IU-EN), we turned to finetuning to see if there was room to improve.

For systems that had been trained on Hansard data concatenated with the first third of the news development data, we experimented with finetuning on just that same first third of news data (using the second third for early stopping and the final third for evaluation), as well as both the first and second thirds (using the final third for both early stopping and evaluation). These approaches improved translation performance in terms of BLEU on the remaining third, with the use of more news data being more effective.[19]

[16] This was the largest size tested; it remains possible that increasing it even more could lead to even better performance.

[17] We also experimented with using novel tags for the domains of the backtranslated data (<PARL> and <EN-NEWS>) with and without additional <BT> tags, but found this had approximately the same effect as combining the backtranslation and domain tags, so we omit it from Table 1.

[18] Additional details of the backtranslation systems and these experiments are in Appendix D.

[19] We expect that training on more of the news data from the start (i.e., two thirds) might improve performance even more, but for our initial experiments we chose to use one third in

Backtranslation Data	Bitext Source Tag	Backtranslation Tag	NH	News.03	Avg
—	—	—	41.0	18.0	29.5
—	<NH|NEWS>	—	**41.3**	18.0	29.7
BT1	—	—	41.0	22.9	32.0
BT1	<NH|NEWS>	<NH|NEWS>	41.0	21.6	31.3
BT1	<NH|NEWS>	<BT> <NH|NEWS>	40.9	23.5	32.2
BT1	<NH|NEWS>	<BT>	40.7	23.8	*32.3*
BT2	—	—	40.8	23.6	32.2
[FINAL] BT2	<NH|NEWS>	<BT>	41.0	25.1	33.1
BT2	—	<BT>	40.9	**26.3**	**33.6**

Table 1: Backtranslation tag experiments on: IU-EN 15k Joint BPE, NH + News.01 (duplicated 15 times), 1.3M EuroParl, 1.3M News. Cased word BLEU scores measured on Hansard (NH) and last third of news (News.03; final 718 lines) portions of newsdev2020-iuen.

We also found that we were able to improve translation of Hansard data by finetuning on recent data. Joanis et al. (2020) observed recency effects when building models with subsets of the data. Here we take that observation a step further and find that finetuning with recent data (Hansard training data from 2017, which was already observed in training) produces BLEU score improvements on Hansard development data on the order of 0.5 BLEU into English, and on the order of 0.7 BLEU into Inuktitut (Tables 2 and 3).[20]

Despite the use of domain tags, finetuning on one domain has negative results for the other (see Tables 2 and 3).

5.6 Ensembling and Hybrid Model

Our hope was to build a single system to translate both news and Hansard but, in the end, we found that our attempts at finetuning for the combination of news and Hansard were outperformed by systems finetuned to one specific domain. Maintaining a held-out third of news data allowed us to measure performance of ensembled models on news data, so long as we only ensembled systems that had not trained on that held-out data. In order to create our final submissions, we chose finetuned systems based on the held-out third, and then ensembled them with the assumption that the strong ensemble with access to the full news development data would outperform the individual systems or pairs of systems trained on subsets. In general, we found that ensembling several systems (using Sockeye's built-in ensembling settings) improved performance. However, this had some limits: for EN-IU if we combined a strong news system whose performance on Hansard had degraded too much with a strong Hansard system whose performance on news had degraded, the final result would be poor performance on both domains.

Our solution to this was simple: decode news data with an ensemble of models finetuned on news, and decode Hansard data with an ensemble of models finetuned on Hansard. Our final submissions are hybrids of domain-specific systems.[21]

System	NH Dev.	ND 3	NH Test	News Test	Full Test
Base: NH+ND.1	24.7	11.7	16.7	11.6	14.1
Base: NH+ND.2	24.7	11.3	16.7	11.2	13.9
Base: NH+ND.3	24.7	–	16.9	12.2	14.5
Ensemble	25.0	–	17.1	13.3	15.1
F.t. ND.{1,2}	21.5	**13.5**	15.0	12.2	13.8
F.t. ND.{2,3}	21.9	–	15.0	13.2	14.4
F.t. ND.{3,1}	20.9	–	13.8	13.1	13.7
Ens.: F.t. ND	21.7	–	14.9	**14.1**	14.8
F.t. NH (from 1)	25.4	11.9	16.9	11.3	14.0
F.t. NH (from 2)	25.4	11.0	16.8	11.0	13.9
F.t. NH (from 3)	25.3	–	16.8	11.3	14.0
Ens.: F.t. NH	**25.7**	–	**17.5**	12.9	15.1
Final hybrid	**25.7**	–	**17.5**	**14.1**	**15.8**

Table 2: BLEU scores of 10k joint BPE EN-IU systems. The best performer is in **bold**. ND=News dev., indexed by thirds. F.t.=Finetuning. Dashes mean a score should not be computed due to test/training data overlap.

6 Submitted Systems

6.1 English–Inuktitut

Our primary submission for EN-IU is a hybrid of two joint BPE 10k ensembled systems with

order to enable us to measure improvements on a held-out set; see Section 5.6 for our efforts to use ensembling to balance the usefulness of training on more data with the ability to measure progress during preliminary experiments.

[20] Note that there is a fine distinction between the two settings here: when finetuning on recent Hansard data, the system is training on data it has already seen. When finetuning on news data, we expose the system to some data it has already seen (one third of the news data) and some data that it has *not* trained on (another third of the news data).

[21] This leaves questions open, e.g., if a Hansard system trained without any news data would perform as well or better on Hansard test data than one trained with news data.

System	NH Dev.	ND 3	NH Test	News Test	Full Test
BT1:NH+ND.1	40.7	23.8	29.0	21.6	25.6
BT2:NH+ND.1	41.0	25.1	29.3	22.1	25.9
BT2:NH+ND.2	41.1	25.1	28.9	22.9	26.1
BT2:NH+ND.3	41.1	–	28.7	22.6	25.9
Ensemble	41.7	–	29.6	24.8	27.4
F.t. ND.{1,2}	39.9	**26.7**	28.5	23.9	26.4
F.t. ND.{2,3}	39.6	–	28.2	23.8	26.1
F.t. ND.{3,1}	40.1	–	28.4	23.7	26.2
Ens.: F.t. ND	40.9	–	29.1	**25.8**	27.6
F.t. NH (from 1)	41.6	23.6	29.0	21.0	25.3
F.t. NH (from 2)	41.5	24.6	28.9	22.8	26.1
F.t. NH (from 3)	41.5	–	28.8	21.6	25.5
Ens.: F.t. NH	**42.4**	–	**29.9**	24.3	27.3
Final hybrid	**42.4**	–	**29.9**	**25.8**	**28.0**

Table 3: BLEU scores of IU-EN systems. The best performer is in **bold** font. ND=News dev., indexed by thirds. F.t.=Finetuning. Dashes mean a score should not be computed due to test/training data overlap.

domain tags. To translate the Nunavut Hansard data, we used an ensemble of three systems, all finetuned on 2017 Hansard data using only the Hansard development data for validation during finetuning. The three base systems used for finetuning were trained on the full Hansard along with the first, second, or third news third (duplicated 15 times), respectively, with BPE-dropout on both the source and target sides.

To translate the news data, we again used an ensemble of three base systems trained with BPE-dropout on both the source and target sides: a base system trained on all Hansard data with the first third of news data (duplicated 15 times) finetuned on the first and second third of news data, another such base system trained instead with the second third of news data (duplicated 15 times) and finetuned on the second and third third of news data, and a final base system trained with the third third of news data (duplicated 15 times) and finetuned on the first and third thirds. The hybrid system had a BLEU score of 15.8 on the test data (Table 2).

6.2 Inuktitut–English

Our primary submission for IU-EN is a hybrid of two joint BPE 15k ensembled systems with domain tags (for news and Hansard bitext) and backtranslation tags (for the backtranslated data). Due to time constraints, we did not run BPE-dropout. Like the EN-IU direction, we built three baseline systems. All baseline systems were trained on the full Hansard training data, along with 1.3 million lines of backtranslated Europarl data and 1.3 million lines of backtranslated news 2019 data. The three baseline systems differed in which third of news was used for training, as described for EN-IU. Backtranslation was performed using an ensemble of the three baseline systems used for the EN-IU task (joint BPE 10k, BPE-dropout).

We performed finetuning on news and recent Hansard in the same manner as for EN-IU. The news test data was translated with the ensemble of news-finetuned systems, while the Hansard test data was translated with the ensemble of the Hansard-finetuned systems. The final system had a BLEU score of 28.0 on the test data (Table 3).

7 Conclusions and Future Work

We have presented the results of our IU-EN and EN-IU systems, showing that a combination of BPE-dropout (for EN-IU), backtranslation (for IU-EN), domain-specific finetuning, ensembling, and hybrid systems produced competitive results. We performed automatic evaluations of the submitted systems in terms of BLEU, chrF (Popović, 2015), and YiSi-1 (Lo, 2020). Our EN-IU system performed best out of the constrained systems in terms of BLEU (15.8, +2.5 above the next-best system), chrF (37.9, +1.7 above the next-best), and YiSi (82.4, +0.5 above the next-best). Our IU-EN system performed third-best out of all systems in terms of BLEU (28.0, -1.9 below the best system), third-best in terms of chrF (48.9, -2.0 below the best system), and third-best in terms of YiSi-1 (92.3, -0.6 behind the best system).[22]

There remains a wide range of future work to be done to improve translation for this language pair. There is still space to improve Roman text output in Inuktitut, perhaps even as simply as an automatic postediting approach. Different subword segmentations (or ones complementary to BPE-dropout like He et al. (2020)), particularly ones that capture morphological and phonological aspects of Inuktitut may also be promising.

In terms of adding monolingual data, we expect that improved data selection for backtranslated data (i.e., to increase topic relevance) may be useful, as would additional Inuktitut monolingual data. Due to time constraints, we were unable to complete BPE-dropout for IU-EN systems; we expect this would have resulted in improved performance.

[22] We do not have information about whether any of these systems were unconstrained. It is also worth noting that the highest-ranked systems differed depending on the metric used, so we await human evaluation.

Domain finetuning remains a challenge given the very small amount of parallel news data available. We did experiment with mixing Hansard and news data for finetuning, but, contrary to Chu et al. (2017), were unable to outperform news-only systems on news. It may be worth trying approaches designed to prevent catastrophic forgetting in domain adaptation (Thompson et al., 2019).

The real test, of course, will be human evaluation; are the systems producing output that might be usable, whether for computer aided translation (via postediting or interactive translation prediction) or for use in other downstream applications?

Acknowledgments

We thank the reviewers for their comments and suggestions. We thank Eddie Santos, Gabriel Bernier-Colborne, Eric Joanis, Delaney Lothian, and Caroline Running Wolf for their comments and feedback on the paper. We thank Chi-kiu Lo for providing automatic evaluation scores of submitted systems. We thank the language experts at Pirurvik Centre for their work on the forthcoming human annotations, and the Government of Nunavut and Nunatsiaq News for providing and allowing the use and processing of their data in this shared task.

References

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. 2011. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 355–362, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Isaac Caswell, Ciprian Chelba, and David Grangier. 2019. Tagged back-translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 53–63, Florence, Italy. Association for Computational Linguistics.

Chenhui Chu, Raj Dabre, and Sadao Kurohashi. 2017. An empirical comparison of domain adaptation methods for neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 385–391, Vancouver, Canada. Association for Computational Linguistics.

Shuoyang Ding, Adithya Renduchintala, and Kevin Duh. 2019. A call for prudent choice of subword merge operations in neural machine translation. In *Proceedings of Machine Translation Summit XVII Volume 1: Research Track*, pages 204–213, Dublin, Ireland. European Association for Machine Translation.

Benoît Farley. 2009. Uqailaut. `www.inuktitutcomputing.ca/Uqailaut/info.php`.

Government of Nunavut. 2020. We speak Inuktut. `https://www.gov.nu.ca/culture-and-heritage/information/we-speak-inuktut`. Accessed August 11, 2020.

Xuanli He, Gholamreza Haffari, and Mohammad Norouzi. 2020. Dynamic programming encoding for subword segmentation in neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3042–3051, Online. Association for Computational Linguistics.

Felix Hieber, Tobias Domhan, Michael Denkowski, David Vilar, Artem Sokolov, Ann Clifton, and Matt Post. 2018. The sockeye neural machine translation toolkit at AMTA 2018. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Papers)*, pages 200–207, Boston, MA. Association for Machine Translation in the Americas.

Inuit Tapiriit Kanatami. 2018. National Inuit Strategy on Research.

Inuit Tapiriit Kanatami. 2020. Unification of the Inuit language writing system.

Eric Joanis, Rebecca Knowles, Roland Kuhn, Samuel Larkin, Patrick Littell, Chi-kiu Lo, Darlene Stewart, and Jeffrey Micher. 2020. The Nunavut Hansard Inuktitut–English parallel corpus 3.0 with preliminary machine translation results. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 2562–2572, Marseille, France. European Language Resources Association.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Rebecca Knowles and Philipp Koehn. 2018. Context and copying in neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3034–3041, Brussels, Belgium. Association for Computational Linguistics.

Catherine Kobus, Josep Crego, and Jean Senellart. 2017. Domain control for neural machine translation. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 372–378, Varna, Bulgaria. INCOMA Ltd.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. *Proceedings of the 10th Machine Translation Summit (MT Summit)*, pages 79–86.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia. Association for Computational Linguistics.

Philippe Langlais, Fabrizio Gotti, and Guihong Cao. 2005. NUKTI: English-Inuktitut word alignment system description. In *Proceedings of the ACL Workshop on Building and Using Parallel Texts*, pages 75–78, Ann Arbor, Michigan. Association for Computational Linguistics.

Samuel Larkin, Boxing Chen, George Foster, Ulrich Germann, Eric Joanis, Howard Johnson, and Roland Kuhn. 2010. Lessons from NRC's Portage system at WMT 2010. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, WMT '10, pages 127–132, Stroudsburg, PA, USA. Association for Computational Linguistics.

Legislative Assembly of Nunavut. 2020. FAQS: What is Hansard? `https://assembly.nu.ca/faq#n125`. Accessed August 11, 2020.

Jason Edward Lewis, Angie Abdilla, Noelani Arista, Kaipulaumakaniolono Baker, Scott Benesiinaabandan, Michelle Brown, Melanie Cheung, Meredith Coleman, Ashley Cordes, Joel Davison, Kūpono Duncan, Sergio Garzon, D. Fox Harrell, Peter-Lucas Jones, Kekuhi Kealiikanakaoleohaililani, Megan Kelleher, Suzanne Kite, Olin Lagon, Jason Leigh, Maroussia Levesque, Keoni Mahelona, Caleb Moses, Isaac ('Ika'aka) Nahuewai, Kari Noe, Danielle Olson, 'Ōiwi Parker Jones, Caroline Running Wolf, Michael Running Wolf, Marlee Silva, Skawennati Fragnito, and Hēmi Whaanga. 2020. Indigenous protocol and artificial intelligence position paper. Project Report 10.11573/spectrum.library.concordia.ca.00986506, Aboriginal Territories in Cyberspace, Honolulu, HI. Edited by Jason Edward Lewis.

Patrick Littell, Anna Kazantseva, Roland Kuhn, Aidan Pine, Antti Arppe, Christopher Cox, and Marie-Odile Junker. 2018. Indigenous language technologies in canada: Assessment, challenges, and successes. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2620–2632, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Chi-kiu Lo. 2020. Extended study of using pretrained language models and YiSi-1 on machine translation evaluation. In *Proceedings of the Fifth Conference on Machine Translation: Shared Task Papers*.

Mick Mallon. 2000. Inuktitut linguistics for technocrats. Ittukuluuk Language Programs. `https://www.inuktitutcomputing.ca/Technocrats/ILFT.php`.

Benjamin Marie, Raphael Rubino, and Atsushi Fujita. 2020. Tagged back-translation revisited: Why does it really work? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5990–5997, Online. Association for Computational Linguistics.

Joel Martin, Howard Johnson, Benoit Farley, and Anna Maclachlan. 2003. Aligning and using an English-Inuktitut parallel corpus. In *Proceedings of the HLT-NAACL 2003 Workshop on Building and Using Parallel Texts: Data Driven Machine Translation and Beyond*, pages 115–118.

Joel Martin, Rada Mihalcea, and Ted Pedersen. 2005. Word alignment for languages with scarce resources. In *Proceedings of the ACL Workshop on Building and Using Parallel Texts*, pages 65–74, Ann Arbor, Michigan. Association for Computational Linguistics.

Jeffrey Micher. 2017. Improving coverage of an Inuktitut morphological analyzer using a segmental recurrent neural network. In *Proceedings of the 2nd Workshop on the Use of Computational Methods in the Study of Endangered Languages*, pages 101–106, Honolulu. Association for Computational Linguistics.

Jeffrey Micher. 2018. Using the Nunavut hansard data for experiments in morphological analysis and machine translation. In *Proceedings of the Workshop on Computational Modeling of Polysynthetic Languages*, pages 65–72, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Carla Parra Escartín, Wessel Reijers, Teresa Lynn, Joss Moorkens, Andy Way, and Chao-Hong Liu. 2017. Ethical considerations in NLP shared tasks. In *Proceedings of the First ACL Workshop on Ethics in Natural Language Processing*, pages 66–73, Valencia, Spain. Association for Computational Linguistics.

Maja Popović. 2015. chrF: character n-gram f-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

Ivan Provilkov, Dmitrii Emelianenko, and Elena Voita. 2020. BPE-dropout: Simple and effective subword regularization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1882–1892, Online. Association for Computational Linguistics.

Lane Schwartz, Francis Tyers, Lori Levin, Christo Kirov, Patrick Littell, Chi kiu Lo, Emily Prud'hommeaux, Hyunji Hayley Park, Kenneth Steimel, Rebecca Knowles, Jeffrey Micher, Lonny Strunk, Han Liu, Coleman Haley, Katherine J. Zhang, Robbie Jimmerson, Vasilisa Andriyanets, Aldrian Obaja Muis, Naoki Otani, Jong Hyuk Park, and Zhisong Zhang. 2020. Neural polysynthetic language modelling.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Controlling politeness in neural machine translation via side constraints. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 35–40, San Diego, California. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016c. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Brian Thompson, Jeremy Gwinnup, Huda Khayrallah, Kevin Duh, and Philipp Koehn. 2019. Overcoming catastrophic forgetting during domain adaptation of neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2062–2068, Minneapolis, Minnesota. Association for Computational Linguistics.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, pages 2214–2218, Istanbul, Turkey. European Languages Resources Association (ELRA).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 5998–6008.

Yining Wang, Jiajun Zhang, Feifei Zhai, Jingfang Xu, and Chengqing Zong. 2018. Three strategies to improve one-to-many multilingual translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2955–2960, Brussels, Belgium. Association for Computational Linguistics.

Marlies van der Wees, Arianna Bisazza, and Christof Monz. 2017. Dynamic data selection for neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1400–1410, Copenhagen, Denmark. Association for Computational Linguistics.

Hayahide Yamagishi, Shin Kanouchi, Takayuki Sato, and Mamoru Komachi. 2016. Controlling the voice of a sentence in Japanese-to-English neural machine translation. In *Proceedings of the 3rd Workshop on Asian Translation (WAT2016)*, pages 203–210, Osaka, Japan. The COLING 2016 Organizing Committee.

A Data and BPE Sizes

For reproducibility, we show the data sizes after cleaning in Table 4. Exact sizes for BPE vocabularies (including codes, and extracted vocabulary sizes) are shown in Table 5.

For IU-EN, we tried disjoint (Inuktitut, English) models with BPE size pairs: (500,500), **(1k,1k)**, (5k,5k), (10k,10k), (500,5k), (1k,5k), and (5k,10k). For EN-IU, we tried disjoint (English, Inuktitut) models with BPE size pairs: (500,500), (1k,1k), (5k,5k), (10k,10k), **(5k,1k)**, (10k,1k), (10k,5k). The typological differences between English and Inuktitut motivated these experiments on vocabulary sizes, testing whether unbalanced sizes may be better than balanced sizes when using disjoint vocabularies. Our intuition was that smaller vocabulary sizes for Inuktitut and larger vocabulary sizes for English might lead to mappings between vocabulary items that were closer to one-to-one.

We also tried using SentencePiece Unigram models (Kudo, 2018) trained for the same size pairs, but they did not yield improvements in BLEU score, so we used `subword-nmt` for the remainder of our experiments.

B Roman Text in Inuktitut Output

When Roman text appears in the Inuktitut side of the Hansard, that same text almost always appears in the English side of the data. In the test data, this is the case for all but 9 of the 151 Roman tokens on the Inuktitut side of the data (over 94%).[23] Thus we expect a good NMT system should learn to produce Roman output that matches some portion of the Roman text in the English input (or at least, that should be true for *most* Roman text it produces).

When the BPE vocabulary model is learned jointly, this task has the potential to be trivial: the system simply needs to learn which source vocabulary tokens are the same as which target vocabulary tokens and then copy them, and RNN-based NMT systems are known to be able to use context in order to do this (Knowles and Koehn, 2018). When the BPE vocabularies are learned independently, however, such words may be broken into different sequences of subwords on the source and target sides, a more challenging task for the system to handle. Intuitively, the system must learn to spell to do this successfully. This led us to experiment with joint vocabulary models. We measured precision, recall, and F1 averaged across Hansard and news development data for Roman words (whole words, after de-BPE, but prior to detokenization) for systems trained using disjoint vocabulary models and those with vocabularies trained jointly (but extracted separately). We found comparable BLEU scores between the two settings, but found that average F1 was higher in the joint setting.

In Table 6, we see three systems from an early set of experiments with identical BLEU scores (the best out of their respective vocabulary size and data balancing sweeps; the disjoint system had the news data repeated 30 times, while the 2k joint system had it repeated 10 times and the 10k joint system had it repeated 30 times). The joint systems had higher F1 scores, particularly driven by improvements in precision (on Hansard, an increase from 30.9% precision to 38.7% and 35.7% and on news an increase from 28.1% to 49.2% and 43.2%).

Nevertheless, as evidenced by the relatively low F1 scores, there best systems still make some errors, as shown in the example below:

Src: Mr. Speaker, the Indspire Awards represent the highest honour the indigenous community presents to its people.

Ref: ᐅᖅᑲᖅᑎ, ᑕᒃᑯᐊ ᐱᑕᖓᖅᑕᖅᑐᑦ ᑭᓪᓗᖅᑐᐃᓂᖅ >ᖅᑐᓂᖅᐸᒥᒃ, ᐅᐱᐟᓇᖅᑐᒥᐅᑕᑯᓗᐃᑦ ᑕᕐᕋᖏᓪ ᓄᓇᖅᑲᖅᓱᖅᑐᙵᔪᓂᖅᓄᑦ ᐃᓄᖅᑯᑎᖏᓐᓄᑦ.

MT: ᐅᖅᑲᖅᑎ, ᑕᒃᑯᐊ Inspiration Award-ᖏᔪᓂᖅᑕᐅᕐᒃ ᖅᑯᑦᑎᓂᖅᐸᒥᒃ ᓄᓇᖅᑲᖅᓱᖅᑐᒥᓂᖅᓄᑦ ᐃᓄᖅᑯᑎᒥᓄᑦ. ᑐᓂᓯᖅᑕᑦᒪᑕ

There are in fact two errors in this MT output: first, the system generates Roman text that it perhaps ought not to have generated, and second it does not successfully copy *Indspire*, instead producing *Inspiration*. This suggests that, although using joint BPE has improved Roman text output in Inuktitut, there is still room for additional improvement. Our final submission had an F1 score of 27.4 (41.3% precision and 20.5% recall)

[23] Of the 9 exceptions, 6 were cases where one side used an abbreviation and the other expanded it, 1 was a plural/singular distinction, 1 was a capitalization difference, and 1 was a spelling difference.

Data set	Sentences	IU words	EN words
Nunavut Hansard 3.0	1299349	7992376	17164079
Nunavut Hansard 3.0 (2017 only)	40951	275248	582480
News (from newsdev2019-eniu)	2156	24980	44507
EN Europarl v10 (full)	2295044		56029587
EN Europarl v10 (subselect)	1300000		31750842
EN News.2019 (full)	33600797		836569124
EN News.2019 (subselect)	1300000		32380145

Table 4: Dataset sizes (post cleaning) of data used in our experiments. Of the monolingual data, only the subselection was used, not the full dataset.

BPE model	Codes	IU Vocab	EN Vocab
IU 1k	573	1006	
EN 1k	794		995
IU 2k	1573	2000	
EN 2k	1794		1978
IU 5k	4573	4991	
EN 5k	4794		4895
IU 10k	9573	9989	
EN 10k	9794		9749
JNT 1k	557	977	519
JNT 2k	1557	1919	1086
JNT 5k	4557	4480	2754
JNT 10k	9557	8597	5071
JNT 15k	12590	12590	7038

Table 5: BPE codes and extracted vocabulary sizes using `subword-nmt` with the `--total-symbols` flag. Single language BPE models are indicated by ISO code and joint models by *JNT*.

System	BLEU	Ave. F1
Disjoint BPE: IU 1k, EN 5k	24.7	24.2
Joint BPE 2k	24.7	25.9
Joint BPE 10k	24.7	27.6

Table 6: Comparison of best disjoint and joint BPE systems trained using Nunavut Hansard and half of the news data as training, scored with BLEU and with Roman text F1 averaged over the Hansard development data and the other half of the news development data. These were early systems trained without tags or back-translation.

C Preprocessing and Postprocessing

In this appendix, we provide detail about our additional language-specific preprocessing and postprocessing.

C.1 Preprocessing

Our additional preprocessing focuses on quotation marks, apostrophes, and some other punctuation. We first describe English-specific preprocessing.

We normalize double quotation marks to three distinct special tokens, `-LDQ-`, `-RDQ-`, and `-UDQ-` (left, right, and unknown double quote, respectively), separated from any surrounding characters by a space. For directional quotation marks ('LEFT DOUBLE QUOTATION MARK' (U+201C) and 'RIGHT DOUBLE QUOTATION MARK' (U+201D)), this is a simple substitution. For straight quotations ('QUOTATION MARK' (U+0022)), we apply the following heuristics: those followed by a space are right, those preceded by a space are left, those followed by punctuation (period, comma, question mark, semicolon) are right, those at the beginning of a line are left, those at the end of a line are right. All that remain are considered unknown.

For single quotes or apostrophes ('LEFT SINGLE QUOTATION MARK' (U+2018) and 'RIGHT SINGLE QUOTATION MARK' (U+2019)), we do as follows. We first convert any instances of 'GRAVE ACCENT' (U+0060) to the right single quote (this is rare but manual examination of the training data suggests that they are used as apostrophes). We then convert any instances of left and right single quotation marks to special (space-separated) tokens `-LSA-` and `-RSA-`, respectively. We next consider 'APOSTROPHE' (U+0027). That token followed by a space is mapped to `-RSA-`, while any instances preceded by a space are mapped to `-LSA-`. Any that are sandwiched between alphanumeric characters (a-z, A-Z, 0-9) are treated as a word internal apostrophe, `-RSI-`. Remaining ones preceded by alphanumeric characters are mapped to `-RSA-`, while those followed by alphanumeric characters are mapped to `-LSA-`. Any remaining at this point are mapped to `-ASO-` (other).

We also map 'EN DASH' (U+2013) to -NDA- and 'EM DASH' (U+2014) to -MDA- (as ever, keeping these special tokens space-separated from remaining text).

For Inuktitut, we use similar substitutions, noting the differences below. This is run after the spelling normalization script provided. For quotation marks, any instances of 'LEFT SINGLE QUOTATION MARK' (U+2018) followed immediately by 'RIGHT SINGLE QUOTATION MARK' (U+2019) are treated as -LDQ-, while any instances of two 'RIGHT SINGLE QUOTATION MARK' (U+2019) in a row are treated as -RDQ-. Double apostrophe is first mapped to 'QUOTATION MARK' (U+0022). Those straight double quotes preceded *or* followed by punctuation (period, comma, question mark, semicolon) are treated as -RDQ-. We expand the earlier alphanumeric matching to include the unicode character range 1400-167F, which contains all syllabics present in the data.

There are five observed types of single quotes or apostrophes in the data. The most common is 'RIGHT SINGLE QUOTATION MARK' (U+2019), appearing more than 9000 times, followed by 'APOSTROPHE' (U+0027), appearing more than 1300 times, followed by 'GRAVE ACCENT' (U+0060), over 600 times, 'LEFT SINGLE QUOTATION MARK' (U+2018), which appears fewer than 200 times, and 'ACUTE ACCENT' (U+00B4), which appears very rarely. We first map the grave accent to 'RIGHT SINGLE QUOTATION MARK' (U+2019). Then, for the remaining four types, if they appear within syllabics (range U+1400 to U+167F), we map them to 'MODIFIER LETTER APOSTROPHE' (U+02BC). This is important because this is then treated as a *non-breaking* character for the purposes of Moses tokenization. It often represents a glottal stop, which *should* be treated as part of the word, not necessarily as something to split on. When one of the four types appears at the end of a word, it is treated as a -RSA- if a left single apostrophe was observed before it in the sentence. Any remaining at the ends of syllabic words are treated as modifier letter apostrophe. Any of the four that appear between non-syllabic alphanumeric characters are mapped to -RSI-. Remaining left single quotation marks are mapped to -LSA-, while remaining right single quotations and acute accents are mapped to -RSA-. Apostrophes are then mapped in the same manner as English, with the addition of the syllabic range to the alphanumeric range.

C.2 Postprocessing

The postprocessing is done to revert the placeholder tokens to appropriate characters and is done after de-BPE-ing and Moses detokenization.

For English, we do as follows. The placeholder -LDQ- and any spaces to the right of it are replaced with 'LEFT DOUBLE QUOTATION MARK' (U+201C), while -RDQ- and any spaces to the left of it are replaced with 'RIGHT DOUBLE QUOTATION MARK' (U+201D), and -UDQ- is replaced with 'QUOTATION MARK' (U+0022) with no modification to spaces.

The -RSI- token and any surrounding spaces are replaced with 'RIGHT SINGLE QUOTATION MARK' (U+2019), -RSA- and any spaces preceding it are replaced with 'RIGHT SINGLE QUOTATION MARK' (U+2019), -LSA- and any spaces following it are replaced with 'LEFT SINGLE QUOTATION MARK' (U+2018), and -ASO- is replaced with 'APOSTROPHE' (U+0027).

The em-dash placeholder is replaced with an em-dash without surrounding spaces, while the en-dash placeholder is replaced with an en-dash *with* surrounding spaces. We also perform some other small modifications to match the most common forms in the original text: spaces around dashes and forward slashes are removed, times are reformatted (spaces removed between numbers with colons and other numbers), space between greater than signs is removed, space is removed before asterisks, spaces are removed following a closing parenthesis that follows a number, three periods in a row are replaced with 'HORIZONTAL ELLIPSIS' (U+2026), space is removed after asterisk that begins a line, space is removed after the pound sign, and space is removed between a right apostrophe and a lowercase s.

For Inuktitut, the postprocessing is similar, with the following changes/additions: the modifier letter apostrophe is replaced with the 'RIGHT SINGLE QUOTATION MARK' (U+2019), no spaces are placed around the en-dash, and spaces are removed between a close parenthesis followed by an open parenthesis.

D Backtranslation Details

Here we describe details of our backtranslation experiments. The first pass (BT1) employed our strongest English–Inuktitut system at the time, trained on the Nunavut Hansard bitext plus the first third of the news bitext (from newsdev2020-eniu) using 5k joint BPE with BPE-dropout on both source and target. Later, we backtranslated the data a second time (BT2) using a stronger three-way ensemble of systems, each of which was trained on the NH corpus and a different third of the news bitext from newsdev2020-eniu using 10k joint BPE with BPE-dropout on both source and target. This ensemble improved the BLEU score on the NH portion of newsdev2020-eniu by 0.5 BLEU (from 24.5 to 25.0); while we could not measure the improvement of the 3-way ensemble on news data, an ensemble of two of these systems (trained using one of the first two thirds of news) yielded a 1.5 boost in BLEU measured on the final news third (from 12.1 to 13.6) over the system used for the first round. Thus the ensembled system used for this second round of backtranslation was stronger at translating both parliamentary and news data.

With BT1 backtranslated data, positive effects came from ensuring that backtranslated data and true bitext are tagged differently. Tagging the backtranslated source with the exact same domain tags as the parallel data leads to a decrease in performance of 1.7–2.2 BLEU for translating news; it is even worse (by 1.3 BLEU on news) than using no tags at all.

While most round one (BT1) backtranslation tagging methods yielded news data BLEU increases between 0.4 and 0.8 (over not tagging), a larger improvement of $\geq$1.5 BLEU occurred when using our second round of backtranslated data (BT2); notably, the worst system trained using BT2 scores only 0.1 BLEU (average) below the best BT1 system. Our best performance was achieved using BT2 backtranslations with <BT> tags but no domain tagging (for either the parallel or backtranslated source). It outperformed the next best system by 1.2 BLEU on news; unfortunately those experiments did not complete before the deadline. Thus our submitted system used the best available systems at the time for additional finetuning: domain tags on parallel data and <BT> tags on the backtranslated data.

Each of the individual systems that contributed to our final Inuktitut–English system combination used 1.3 million lines of Europarl (tagged as <BT>), 1.3 million lines of news (tagged as <BT>), approximately 1.3 million lines of Nunavut Hansard (tagged as <NH>), and 719 or 718 lines of news (tagged as <NEWS> and duplicated 15 times).

E Inuktitut Common Crawl and Additional Data

We did not use any data from Inuktitut Common Crawl in our submissions. In our initial experiments, we found it generally harmed translation quality. Nevertheless, we provide here some observations from our analysis, in the hopes that they are useful to other researchers. First, the Common Crawl data provided contains fairly large amounts of non-Inuktitut data. This includes noise, such as long sequences of characters (like lists of characters) as well as text art (such as English words spelled using visually similar syllabics and other characters, e.g., ᕼᕮᒪᒪᑐ). There is also text in several other languages and dialects, including, but likely not limited to: ᓇᓴᐲᐱᐣ (Naskapi),[24] ᓂᐦᐃᔨᐁᐧᐃᐦᐣ (Plains Cree), and ᓄᓇᕗᑦᑎᐅᑎᑐᑦ (Nunavimmiutitut, an Inuit language spoken in Nunavik).[25] Of particular note is the latter, which – while it is the only one of the three within the Inuit dialect continuum that includes Inuktitut – is an Inuit language (sometimes called the Nunavik dialect of Inuktitut) that makes use of one additional column in the syllabary (∇, ∨, ∪, ..., or *ai*, *pai*, *tai*, ...). Those characters do not appear in the Hansard, thus rendering it impossible for our systems to translate them exactly without some form of modification, even if they might otherwise share similarities with words that appear in the Hansard. Removing characters that were not observed in the Hansard data (which helps filter out some non-Inuktitut language data) and filtering out potential text art results in a much smaller Common Crawl data set, less than half the size of the original.

While additional monolingual or bilingual data would likely benefit English to Inuktitut translation, we encourage non-Inuit researchers who plan to perform data collection to do so in collaboration with Inuit communities and language speakers.

[24] Text appears to be scraped from the Naskapi Community Web Site, http://www.naskapi.ca/.

[25] https://www.kativik.qc.ca/our-schools/resources/

The efforts of Inuit language experts at Pirurvik Centre were vital to the analysis of the data used for this task (Joanis et al., 2020), collected through communications with Nunatsiaq News and the Government of Nunavut with the goal of selecting data usable for this translation task, both in terms of public availability and language. Aside from the machine learning related risks of accidentally collecting data from other languages and labeling it as Inuktitut (as we observed in the Common Crawl data), there are also ethical concerns. While it does not focus specifically on language data, the National Inuit Strategy on Research (NISR, Inuit Tapiriit Kanatami, 2018) highlights as a priority "Ensuring Inuit access, ownership and control over data and information" and focuses on partnership with Inuit organizations, transparency, and data sharing to end exploitative research practices and build research relationships that respect Inuit self-determination.[26] The NISR contains a discussion of potential harms of research done without relationships to the communities impacted by it, with both Inuit-specific concerns and concerns from a broader history of colonialism. Lewis et al. (2020) provide a discussion of guidelines for Indigenous-centred AI from a variety of Indigenous perspectives (though not specifically from Inuit perspectives), including topics of ethics, data sovereignty, and responsibility and relationships in AI. Building and maintaining community relationships and collaborations can help ensure that data is handled and shared in ways that respect cultural values and Indigenous intellectual property,[27] which outsiders may not be familiar with. A full discussion of these topics is beyond the scope of this paper, but we raise the discussion here as part of the process of working towards best practices in building respectful research relationships that centre community goals at all steps of the research process.

F Statement on Avoiding Conflicts of Interest

In their work on ethical considerations in shared tasks, Parra Escartín et al. (2017) raise the issue of actual or perceived conflicts of interest between task organizers and participants. We provide the following information in the interest of transparency.

The data for the shared task on Inuktitut-English was collected by researchers at the National Research Council of Canada (NRC) in collaboration with the Pirurvik Centre.[28] The team of researchers at NRC was divided into two groups: those working on task organization and those participating in the shared task (the latter group are the authors of this paper). In order to prevent unfair advantages to the task participants, the organizers did not discuss the web source or details of the evaluation set with the participants at any time before the submission of the systems.

We did communicate with the organizers to receive clarification regarding the spurious quotes in the test data; the response to this was distributed to the full WMT mailing list.

[26] https://www.itk.ca/wp-content/uploads/2018/04/ITK_NISR-Report_English_low_res.pdf

[27] United Nations Declaration on the Rights of Indigenous Peoples, Article 31.

[28] https://www.pirurvik.ca/

CUNI Submission for the Inuktitut Language in WMT News 2020

Tom Kocmi

Charles University, Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics
Malostranské náměstí 25, 118 00 Prague, Czech Republic
kocmi@ufal.mff.cuni.cz

Abstract

This paper describes CUNI submission to the WMT 2020 News Translation Shared Task for the low-resource scenario Inuktitut–English in both translation directions. Our system combines transfer learning from a Czech–English high-resource language pair and backtranslation. We notice surprising behaviour when using synthetic data, which can be possibly attributed to a narrow domain of training and test data. We are using the Transformer model in a constrained submission.

1 Introduction

The rapid development of Neural Machine Translations (NMT) systems helped NMT in approaching human translation quality for high-resource language pairs like Chinese–English (Hassan et al., 2018) or English–Czech (Popel et al., 2020). This is not true for a low-resource scenario, where the lack of a large quantity of parallel data limits the performance of an NMT system. Thus, in recent years the research focused on low-resource NMT become important.

In this paper, we describe our approach to low-resource NMT of Inuktitut–English. We use the standard Transformer-big model (Vaswani et al., 2017) and apply two techniques to improve the performance on the low-resource language, namely transfer learning (Kocmi and Bojar, 2018) and backtranslation (Sennrich et al., 2016). We used a similar approach in WMT19 for Gujarati and Kazakh machine translation (Kocmi and Bojar, 2019).

Training low-resource model solely on authentic parallel data results in poor performance, and that results in low quality of generated backtranslation of monolingual data as well. Hence transfer learning is as an excellent method to first improve the performance of the NMT system that is later used for backtranslation of monolingual data.

2 Background

In this section, we describe the technique of transfer learning, backtranslation, and models that we used for training Inuktitut–English models.

2.1 Transfer Learning

Kocmi and Bojar (2018) presented a method of transfer learning that uses a high-resource language pair to train the "parent" model. After the training convergence, the parent training data are replaced with the training data of the low-resource language pair ("child"). Then the technique of fine-tuning continues without changing any parameters, resetting moments, nor changing the learning rate.

This technique has one shortcoming, and that is the problem with vocabulary mismatch. Kocmi and Bojar (2018) overcome this problem by preparing a shared vocabulary for all languages in both parent and child language pairs in advance. Their approach is to prepare a mixed subword vocabulary from the concatenation of training corpora for both languages.

We use their *balanced vocabulary* approach that combines an equal amount of parallel data from both training corpora, under-sampling the high-resource language pair as needed. Hence the low-resource language subwords are represented in the vocabulary with roughly the same prominence as the high-resource language pair ones.

Kocmi (2019) showed that parent and child language pairs do not have to be linguistically related, and more crucial criterion is the amount of parent parallel data. For this reason, we have selected Czech–English as a parent language pair, because it is one of the most resource-full languages allowed for WMT 2020.

Proceedings of the 5th Conference on Machine Translation (WMT), pages 171–174
Online, November 19–20, 2020.

2.2 Backtranslation

The amount of available monolingual data typically exceeds the amount of available parallel data. One of the techniques for using monolingual data in NMT is called backtranslation (Sennrich et al., 2016). It uses a model trained in the reverse direction to translate monolingual data to the source language of the first model. Backtranslated sentences are then aligned with their monolingual sentences to create synthetic parallel corpora. The standard practice is to mix the authentic parallel corpora with the synthetic ones, although it is not the only possible approach. Popel (2018) obtained better results by repeatedly alternating between the training on the authentic and the synthetic portion of the parallel data instead of mixing them. We got inspired with this approach, and after training on synthetic data, we add a step to fine-tune again with authentic parallel data.

The performance of the backtranslation model is essential. Especially in the low-resource scenario, where the baseline models trained only on the authentic parallel data have a poor score, and they generate very low quality backtranslated data. Therefore, we first improve the performance of baseline with the transfer learning and generate the synthetic data of better quality.

Synthetic data can be noisy; therefore, we remove synthetic sentence pairs containing repetitive patterns, which is often a case of bad translation. We also remove sentences that contain Latin script in Inuktitut translations as Inuktitut has its script. This filtration reduced the number of synthetic sentences from 51.7M to 45.5M sentences.

2.3 Datasets and Model

All our models are trained only on the data allowed for the WMT 2020 News shared task. We use all available parallel data for Inuktitut–English prepared by Joanis et al. (2020). We use Czech–English corpus CzEng20 created by Kocmi et al. (2020) as a parent language pair. This corpus contains 61M authentic parallel sentences and also two sets of synthetic parallel sentences of similar size. For training parent model, we used all authentic parallel data plus one set of synthetic data with authentic English part. We ignored synthetic data with authentic Czech. The reason is that we assume the child model transfers knowledge mainly for the English language that is shared between both parent and child language pair. Additionally, we use monolingual English sentences from News Crawl 2018 and 2019 for backtranslation step. Results in section 3 are computed based on official WMT20 testset for Inuktitut–English (Joanis et al., 2020). All used training data are presented in table 1. We remove empty lines from Inuktitut–English training set.

Lang. pair	Sent. pairs	Words (CS/IU)	Words (EN)
CS–EN	137.2M	1913M	2176M
IU–EN	1.3M	8M	17M
Mono EN	51.7M	-	1756M

Table 1: More details on the training sizes of training corpora. Columns with words show number of words separated by space. All data are from http://statmt.org/wmt20/.

As for the model, we use the Transformer "big single GPU" configuration as described in Vaswani et al. (2017), model which translates through an encoder-decoder with each layer involving an attention network followed by a feed-forward network. We use the version 1.11 of sequence-to-sequence implementation of Transformer called tensor2tensor.[1]

Popel and Bojar (2018) documented best practices to improve the performance of the model. Based on their observation, we use the Adafactor optimizer with inverse square root decay and 16000 warmup steps. Based on our previous experiments (Kocmi et al., 2018), we set the maximum number of subwords in a sentence to 100. The benefit is that the batch size can be increased to 4500 for our GPUs. The experiments are trained on a single GPU NVidia GeForce 1080 Ti or Quadro P5000.

3 Results

All reported results are calculated on the testset of WMT 2020 and evaluated with case sensitive SacreBLEU (Post, 2018).[2] The evaluation is done on unmodified outputs of our system. For final WMT20 submission, we have automatically corrected quotes to match the source. This step is not used for results in table 2.

The baseline models in table 2 are trained on the authentic data only. We have not focused on the backtranslation step for EN→IU as there are only 165k monolingual Inuktitut sentences available.

In IU→EN "Transfer from CS–EN" we get an

[1] https://github.com/tensorflow/tensor2tensor

[2] The SacreBLEU signature is BLEU + case.mixed + numrefs.1 + smooth.exp + tok.13a + version.1.4.6

Training dataset	IU→EN	EN→IU
Authentic (baseline)	20.10	9.52
Transfer from CS–EN	22.98	10.41
Synthetic + auth	20.91	-
Authentic only	25.38	-

Table 2: Test set BLEU scores of our setup. Except for the baseline, each column shows improvements obtained after fine-tuning a model one line up on different datasets.

improvement of almost 3 BLEU. For a model, where we used a mix of "synthetic and authentic" data that is generated by our EN→IU model, we can notice a performance dropped from 22.98 to 20.91. However, following with fine-tuning this model again with authentic data, we get an increase in performance to 25.38. This is unexpected behaviour. Our understanding is that it could be attributed to a narrow domain of train and testset containing mainly speech transcripts. At the same time, the synthetic data are generated from English news articles, which is a more general domain. Therefore, while the model is trained on the general domain, it loses score on a domain-specific testset. This could be tested if we could obtain a testset on a different domain than speech transcription; however, we do not have such testset available.

During training on synthetic data, the model learns a general domain and loses performance on domain-specific testset to 20.91 BLEU, however after fine-tuning again on authentic domain-specific data only it reaches the highest performance of 25.38 BLEU.

4 Conclusion

We participated in the low-resource Inuktitut–English in the WMT 2020 News Translation Shared Task. We combined transfer learning with the back-translation and obtained significant improvements.

Surprisingly, we found out that although training the model on backtranslated data decreases the performance of the system in terms of BLEU score; it is still helpful when continued with fine-tuning on authentic data. We believe this is mainly because the Inuktitut–English training and test data are from a narrow domain of legal texts.

Acknowledgments

This study was supported in parts by the grants 18-24210S of the Czech Science Foundation and 825303 (Bergamot) of the European Union. This work has been using language resources and tools stored and distributed by the LINDAT/CLARIN project of the Ministry of Education, Youth and Sports of the Czech Republic (LM2015071).

References

Hany Hassan, Anthony Aue, Chang Chen, Vishal Chowdhary, Jonathan Clark, Christian Federmann, Xuedong Huang, Marcin Junczys-Dowmunt, William Lewis, Mu Li, Shujie Liu, Tie-Yan Liu, Renqian Luo, Arul Menezes, Tao Qin, Frank Seide, Xu Tan, Fei Tian, Lijun Wu, Shuangzhi Wu, Yingce Xia, Dongdong Zhang, Zhirui Zhang, and Ming Zhou. 2018. Achieving human parity on automatic chinese to english news translation. *CoRR*, abs/1803.05567.

Eric Joanis, Rebecca Knowles, Roland Kuhn, Samuel Larkin, Patrick Littell, Chi-kiu Lo, Darlene Stewart, and Jeffrey Micher. 2020. The nunavut hansard inuktitut–english parallel corpus 3.0 with preliminary machine translation results. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 2562–2572.

Tom Kocmi. 2019. *Exploring Benefits of Transfer Learning in Neural Machine Translation*. Ph.D. thesis, Charles University.

Tom Kocmi and Ondřej Bojar. 2018. Trivial Transfer Learning for Low-Resource Neural Machine Translation. In *Proceedings of the 3rd Conference on Machine Translation (WMT): Research Papers*, Brussels, Belgium.

Tom Kocmi and Ondřej Bojar. 2019. CUNI Submission for Low-Resource Languages in WMT News 2019. In *Proceedings of the Fourth Conference on Machine Translation, Volume 2: Shared Task Papers*, Florence, Italy. Association for Computational Linguistics.

Tom Kocmi, Martin Popel, and Ondrej Bojar. 2020. Announcing czeng 2.0 parallel corpus with over 2 gigawords. *arXiv preprint arXiv:2007.03006*.

Tom Kocmi, Roman Sudarikov, and Ondřej Bojar. 2018. CUNI Submissions in WMT18. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 435–441, Belgium, Brussels. Association for Computational Linguistics.

Martin Popel. 2018. Machine translation using syntactic analysis. *Univerzita Karlova*.

Martin Popel and Ondřej Bojar. 2018. Training Tips for the Transformer Model. *The Prague Bulletin of Mathematical Linguistics*, 110(1):43–70.

Martin Popel, Marketa Tomkova, Jakub Tomek, Łukasz Kaiser, Jakob Uszkoreit, Ondřej Bojar, and Zdeněk Žabokrtskỳ. 2020. Transforming machine translation: a deep learning system reaches news translation quality comparable to human professionals. *Nature communications*, 11(1):1–15.

Matt Post. 2018. A call for clarity in reporting bleu scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 6000–6010. Curran Associates, Inc.

Tilde at WMT 2020: News Task Systems

Rihards Krišlauks[†‡] and **Mārcis Pinnis**[†‡]
[†]Tilde / Vienibas gatve 75A, Riga, Latvia
[‡]Faculty of Computing, University of Latvia / Raiņa bulv. 19, Riga, Latvia
{firstname.lastname}@tilde.lv

Abstract

This paper describes Tilde's submission to the WMT2020 shared task on news translation for both directions of the English↔Polish language pair in both the constrained and the unconstrained tracks. We follow our submissions form the previous years and build our baseline systems to be morphologically motivated sub-word unit-based Transformer base models that we train using the Marian machine translation toolkit. Additionally, we experiment with different parallel and monolingual data selection schemes, as well as sampled back-translation. Our final models are ensembles of Transformer base and Transformer big models which feature right-to-left re-ranking.

1 Introduction

This year, we developed both constrained and unconstrained NMT systems for the English↔Polish language pair. We base our methods on the submissions of the previous years (Pinnis et al., 2017b, 2018, 2019) including methods for parallel data filtering from Pinnis (2018). Specifically, we lean on Pinnis (2018) and Junczys-Dowmunt (2018) for data selection and filtering, (Pinnis et al., 2017b) for morphologically motivated sub-word units and synthetic data generation, Edunov et al. (2018) for sampled back-translation and finally Morishita et al. (2018) for re-ranking with right-to-left models. We use the Marian toolkit (Junczys-Dowmunt et al., 2018) to train models of Transformer architecture (Vaswani et al., 2017).

Although document level NMT as showcased by (Junczys-Dowmunt, 2019) have yielded promising results for the English-German language pair, we were not able to collect sufficient document level data for the English-Polish language pair. As a result, all our systems this year translate individual sentences.

The paper is further structured as follows: Section 2 describes the data used to train our NMT systems, Section 3 describes our efforts to identify the best-performing recipes for training of our final systems, Section 5 summarises the results of our final systems, and Section 6 concludes the paper.

2 Data

For training of the constrained NMT systems, we used data from the WMT 2020 shared task on news translation[1]. For unconstrained systems, we used data from the Tilde Data Library[2]. The 10 largest publicly available datasets that were used to train the unconstrained systems were Open Subtitles from the Opus corpus (Tiedemann, 2016), ParaCrawl (Banón et al., 2020) (although it was discarded due to noise found in the corpus), DGT Translation Memories (Steinberger et al., 2012), Microsoft Translation and User Interface Strings Glossaries[3] from multiple releases up to 2018, the Tilde MODEL corpus (Rozis and Skadiņš, 2017), WikiMatrix (Schwenk et al., 2019), Digital Corpus of the European Parliament (Hajlaoui et al., 2014), JRC-Acquis (Steinberger et al., 2006), Europarl (Koehn, 2005), and the QCRI Educational Domain Corpus (Abdelali et al., 2014).

2.1 Data Filtering and Pre-Processing

First, we filtered data using Tilde's parallel data filtering methods (Pinnis, 2018) that allow discarding sentence pairs that are corrupted, have low content overlap, feature wrong language content, feature too high non-letter ratio, etc. The exact filter configuration is defined in the paper by (Pinnis, 2018).

Then, we pre-processed all data using Tilde's parallel data pre-processing workflow that nor-

[1]http://www.statmt.org/wmt20/translation-task.html
[2]https://www.tilde.com/products-and-services/data-library
[3]https://www.microsoft.com/en-us/language/translations

Proceedings of the 5th Conference on Machine Translation (WMT), pages 175–180
Online, November 19–20, 2020.

Scenario	Lang. pair	Raw	Filtered	
			Tilde	+DCCEF
(c)	En → Pl	10.8M	6.5M	4.3M
	Pl → En			4.3M
(u)	En → Pl	55.4M	31.5M	23.3M
	Pl → En			24.1M
(u) w/o PC	En → Pl	48.8M	27.0M	21.6M
	Pl → En			21.3M

Table 1: Parallel data statistics before and after filtering. (c) - constrained, (u) - unconstrained, "w/o PC" - "without ParaCrawl".

malizes punctuation (quotation marks, apostrophes, decodes HTML entities, etc.), identifies non-translatable entities and replaces them with placeholders (e.g., e-mail addresses, Web site addresses, XML tags, etc.), tokenises the text using Tilde's regular expression-based tokeniser, and applies true-casing.

In preliminary experiments, we identified also that morphology-driven word splitting (Pinnis et al., 2017a) for English↔Polish allowed us to increase translation quality by approximately 1 BLEU point. The finding complies with our findings from previous years (Pinnis et al., 2018, 2017b). Therefore, we applied morphology-driven word splitting also for this year's experiments.

Then, we trained baseline NMT models (see Section 3.2) and language models, which are necessary for dual conditional cross-entropy filtering (DCCEF) (Junczys-Dowmunt, 2018) in order to select parallel data that is more similar to the news domain (for usefulness of DCCEF, refer to Section 3.3). For in-domain (i.e., news) and out-of-domain language model training, we used four monolingual datasets of 3.7M and 10.6M sentences[4] for the constrained and unconstrained scenarios respectively. Once the models were trained, We filtered parallel data using DCCEF. The parallel data statistics before and after filtering are given in Table 1.

For our final systems, we also generated synthetic data by randomly replacing one to three content words on both source and target sides with unknown token identifiers. This has shown to increase robustness of NMT systems when dealing with rare or unknown phenomena (Pinnis et al., 2017a). This process almost doubles the size of the corpora, therefore, this was not done for the datasets that were used for the experiments documented in Section 3.

For backtranslation experiments, we used all available monolingual data from the WMT shared task on news translation. In order to make use of the Polish CommonCrawl corpus, we scored sentences using the in-domain language models and selected top-scoring sentences as additional monolingual data for back-translation.

Many of the data processing steps were sped up via parallelization with GNU Parallel (Tange, 2011).

3 Experiments

In this section, we describe the details of the methods used and experiments performed to identify the best-performing recipes for training of Tilde's NMT systems for the WMT 2020 shared task on news translation. All experiments that are described in this section were carried out on the constrained datasets unless specifically indicated that also unconstrained datasets were used.

3.1 NMT architecture

All NMT systems that are described further have the Transformer architecture (Vaswani et al., 2017). We trained the systems using the Marian toolkit (Junczys-Dowmunt et al., 2018). The Transformer *base* model configuration was used throughout the experiments except for the experiments with the *big* model configuration that are described in Section 5. We used gradient accumulation over multiple physical batches (the *--optimizer-delay* parameter in Marian) to increase the effective batch size to around 1600 sentences in the *base* model experiments and 1000 sentences in *big* model experiments. The Adam optimizer with a learning rate of 0.0005 and with 1600 warm-up update steps (i.e., the learning rate linearly rises during warm-up; afterwards decays proportionally to the inverse of the square root of the step number) was used. For language model training, a learning rate of 0.0003 was used.

3.2 Baseline models

We trained baseline models using the Transformer *base* configuration as defined in Section 3.1. The validation results for the baseline NMT systems are provided in Table 2. As we noticed last year that the ParaCrawl corpus contained a large proportion (by our estimates up to 50%) (Pinnis et al., 2019)

[4]The sizes correspond to the smallest monolingual in-domain dataset, which in both cases were news in Polish. For other datasets, random sub-sets were selected.

System	En → Pl	Pl → En
Constrained		
Baseline	21.67	32.69
+DCCEF	**22.19**	**33.45**
Unconstrained		
Baseline	21.86	**33.08**
+DCCEF	22.51	30.86
Baseline w/o ParaCrawl	**24.29**	29.47
+DCCEF	22.60	28.59

Table 2: Comparison of baseline NMT systems trained on data that were prepared with and without DCCEF.

of machine translated content, we trained baseline systems with and without ParaCrawl. It can be seen that when training the En → Pl unconstrained system using ParaCrawl, we loose over 2 BLEU points. This is because most machine translated content is on the non-English (in this case Polish) side. For the Pl → En direction, the machine-translated content acts as back-translated data and, therefore, does not result in quality degradation. Further, our Pl → En systems are trained using ParaCrawl, and En → Pl systems – without ParaCrawl.

3.3 Dual Conditional Cross-Entropy Filtering

After the baseline systems, we analysed whether DCCEF allows improving translation quality. The validation results in Table 2 show that translation quality increases for the constrained systems, but degrades for the unconstrained systems. Further, we used DCCEF only for the constrained scenario systems.

3.4 Back-translation

We used monolingual data back-translation to adapt the NMT systems to the news domain. Edunov et al. (2018) has shown that using output sampling instead of beam-search during back-translation yields better-performing NMT systems. Hence, we exclusively used output sampling for monolingual data back-translation. However, due to the abundance of monolingual data for both translation directions, we experimented with different rates of upsampling and back-translated data cutoff to determine whether translation performance doesn't degrade in the presence of a too small proportion of bitext data during training.

Another dimension of inquiry was with different strategies for data filtering in the preparation of the back-translated data. Ng et al. (2019) have described a method for domain data extraction from general domain monolingual data using domain and out-of-domain language models. We compared said method with a simpler alternative of using only an in-domain language model for in-domain data scoring. We sorted the monolingual data according to the scores produced by the in-domain language model or by the combination of in-domain and out-of-domain language model scores and experimented with different cutoff points when selecting data for back-translation.

Considering the above, we carried out experiments along two dimensions – 1) monolingual data selection strategy, which was either *combined* or *in-domain*, signifying whether the combined score of both language models or just the score from the in-domain language model was used, respectively, and 2) the bitext and synthetic data mixture selection strategy, which was one of:

- *original ratio* – all available bitext data for the translation direction were combined with all back-translated data having a score ≥ 0, when using the *combined* selection strategy, or N top-scoring back-translated sentences, when using the *in-domain* selection strategy, where N was selected to match the amount of synthetic data selected in the *combined* case.

- *upsampled 1:1* – the same amount of synthetic data was selected as with the *original ratio* data mixture selection strategy, but bitext was upsampled to match the amount of synthetic data.

- *cutoff* {*1:1, 1:2, 1:3*} – all available bitext data for the translation direction were combined with N top-scoring back-translated sentences where N was chosen so that the ratio of bitext to synthetic data was either 1:1, 1:2 or 1:3.

As a result of the above, we ended up with 96.8M sentences (14% retained) from the English monolingual corpus and 137M (99% retained) sentences from the Polish monolingual corpus after applying the *combined* data selection strategy. Consequently, the same amount of data was selected for the *in-domain* data selection strategy in the case of *original ratio* and *upsampled 1:1* data mixture selection strategies (i.e. when not doing *cutoff*).

	orig. ratio	ups. 1:1	cutoff		
			1:1	1:2	1:3
En → Pl					
combined	23.35	24.01	24.52	24.72	-
in-domain	22.10	22.92	25.02	**25.28**	25.24
Pl → En					
combined	31.19	33.45	33.29	**33.60**	-
in-domain	29.67	-	33.40	33.28	-

Table 3: En → Pl back-translation experiment results.

The results for back-translation experiments are presented in Table 3. The systems use the DCCEF-filtered constrained datasets and therefore are directly comparable to the constrained DCCEF systems in Table 2.

For our final systems, we use the *combined* selection strategy for Pl → En and the *in-domain* selection strategy for En → Pl. For unconstrained systems, we identified that there is no significant difference between translation quality; we used the *combined* selection strategy for both language pairs.

3.5 QHAdam optimizer

Last year (Pinnis et al., 2019) we used the QHAdam optimizer (Ma and Yarats, 2018) for model training, however, we hadn't treated QHAdam and Adam the same in the experimental process, having dedicated substantially more effort to optimizer hyperparameter tuning for QHAdam than Adam. To make an unbiased comparison of the two optimizers, we trained corresponding system variants using QHAdam for the *combined cutoff 1:2*, *in-domain cutoff 1:2* and *in-domain cutoff 1:3* systems from Section 3.4 in the En → Pl translation direction. The BLEU scores for the experiments are found in Table 4. We see that QHAdam performs no better than Adam. We had also done more extensive experiments comparing QHAdam to Adam for a range of learning rate and warm-up step parameter settings on a different dataset, which showed a similar trend, however we do not present those results here. As a result, we didn't choose QHAdam over Adam in this year's competition.

We note, however, that we used the recommended safe defaults for the QHAdam's hyperparameters – $v_1 = 0.8, v_2 = 0.7$ – and we haven't performed a search over these values which could have yielded different results.

	combined cutoff 1:2	in-domain cutoff	
		1:2	1:3
Adam	24.72	**25.28**	**25.24**
QHAdam	**24.86**	24.98	25.00

Table 4: BLEU scores for the QHAdam experiments in the En → Pl translation direction.

3.6 Right-to-Left Re-Ranking

Morishita et al. (2018) report improving the translation performance by using right-to-left (R2L) reranking. The method employs a right-to-left model to re-score the n-best list outputs of a regular – left-to-right – model by multiplying both models' translation probabilities. We implement R2L reranking the same as Morishita et al. (2018), but opted to use n-best lists with $n = 12$ (instead of $n = 10$).

The R2L re-ranking experiments were performed during the final stages of the competition, hence the baseline systems for those experiments were the final systems that were being prepared for submission to the news translation task. Therefore we present the results in Table 5 in the Results section. We find similar improvements as Morishita et al. (2018), albeit they are slightly smaller.

4 Final Systems

We chose the best performing system variants from Section 3 to serve as a base for the final submission for the news translation task. For the constrained scenario, we trained final systems using parallel data that were filtered with Tilde's filtering methods and DCCEF, back-translated monolingual data using a ratio of 1:2 (different data selection methods were applied for both translation directions), and synthetic data featuring unknown phenomena. For the unconstrained scenario, we trained final systems using parallel data that were filtered only with Tilde's filtering methods, back-translated monolingual data that were selected using the *combined* data selection strategy using a ratio of 1:1, and synthetic data featuring unknown phenomena. All models were trained using the Adam optimiser.

When preparing the final systems, we also employed R2L re-ranking (see Section 3.6), ensembling of the best three models, and trained Transformer models using the *big* model configuration.

	Constrained	Unconstrained
Pl → En		
Base	33.48	32.63
+R2L	34.34	33.29
Big	33.79	33.15
+R2L	**34.83**	33.45
Ensemble of 3	34.19	33.39
+R2L	34.80	33.53
En → Pl		
Base	25.64	26.12
+R2L	26.24	26.52
Big	25.59	26.47
+R2L	26.70	26.78
Ensemble of 3	26.07	26.86
+R2L	26.73	**27.12**

Table 5: Final system evaluation results (BLEU scores) on validation data (bold marks best scores; submitted systems are underlined).

5 Results

The BLEU scores for the systems that were evaluated for the final submission are shown in Table 5. The results show that right-to-left reranking increased translation quality for all systems. For the En → Pl translation direction, the best results were achieved when using ensembles of three models and better results were achieved by the unconstrained systems. However, for the Pl → En translation direction, the unconstrained systems achieved lower results than the constrained systems. The best results were achieved by the Transformer big model; ensembling did not improve results.

In overall, the results differ from what we have observed in previous years. Back-translation for Pl → En did not improve results, which raises a question of a possible domain mismatch between the monolingual data we back-translated and the development data. Unconstrained systems are only slightly better than constrained systems for En → Pl and even subpar for the Pl → En translation direction, which shows that current NMT methods are not able to benefit from larger datasets. Hence, having in-domain data is more important.

6 Conclusion

In this paper, we described Tilde's NMT systems for the WMT shared task on news translation. This year, we trained constrained and unconstrained systems for the English↔Polish language pair. We detailed the methods applied and the training recipes.

During our experiments, we identified several avenues of possible further research. We saw that larger datasets even after applying data selection methods did not allow improving translation quality (at least not significantly). We made a similar observation also previous years when participating in WMT. We saw in our results also that back-trannslation did not yield positive results for En → Pl. We hypothesise that there may be a domain mismatch between the data we used for training and the newsdev2020 dataset. However, this requires further investigation.

Acknowledgements

The research has been supported by the European Regional Development Fund within the research project "Multilingual Artificial Intelligence Based Human Computer Interaction" No. 1.1.1.1/18/A/148. We thank the High Performance Computing Center of Riga Technical University for providing access to their GPU computing infrastructure.

References

Ahmed Abdelali, Francisco Guzman, Hassan Sajjad, and Stephan Vogel. 2014. The amara corpus: Building parallel language resources for the educational domain. In *LREC*, volume 14, pages 1044–1054.

Marta Banón, Pinzhen Chen, Barry Haddow, Kenneth Heafield, Hieu Hoang, Miquel Espla-Gomis, Mikel L Forcada, Amir Kamran, Faheem Kirefu, Philipp Koehn, et al. 2020. Paracrawl: Web-scale acquisition of parallel corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4555–4567.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding Back-Translation at Scale. *arXiv:1808.09381 [cs]*. ArXiv: 1808.09381.

Najeh Hajlaoui, David Kolovratnik, Jaakko Väyrynen, Ralf Steinberger, and Dániel Varga. 2014. Dcep-digital corpus of the european parliament. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*.

Marcin Junczys-Dowmunt. 2018. Dual conditional cross-entropy filtering of noisy parallel corpora. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 888–895.

Marcin Junczys-Dowmunt. 2019. Microsoft translator at wmt 2019: Towards large-scale document-level neural machine translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 225–233.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast Neural Machine Translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia. Association for Computational Linguistics.

Philipp Koehn. 2005. Europarl : A Parallel Corpus for Statistical Machine Translation. In *Proceedings of the 10th Machine Translation Summit (MT Summit)*, pages 79–86.

Jerry Ma and Denis Yarats. 2018. Quasi-Hyperbolic Momentum and Adam for Deep Learning. *arXiv preprint arXiv:1810.06801*.

Makoto Morishita, Jun Suzuki, and Masaaki Nagata. 2018. NTT's Neural Machine Translation Systems for WMT 2018. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 461–466, Belgium, Brussels. Association for Computational Linguistics.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook FAIR's WMT19 News Translation Task Submission. *arXiv:1907.06616 [cs]*. ArXiv: 1907.06616.

Mārcis Pinnis. 2018. Tilde's Parallel Corpus Filtering Methods for WMT 2018. In *Proceedings of the Third Conference on Machine Translation*, pages 952–958, Brussels, Belgium. Association for Computational Linguistics.

Mārcis Pinnis, Rihards Krišlauks, Daiga Deksne, and Toms Miks. 2017a. Neural Machine Translation for Morphologically Rich Languages with Improved Sub-word Units and Synthetic Data. In *Proceedings of the 20th International Conference of Text, Speech and Dialogue (TSD2017)*, volume 10415 LNAI, Prague, Czechia.

Mārcis Pinnis, Rihards Krišlauks, Toms Miks, Daiga Deksne, and Valters Šics. 2017b. Tilde's Machine Translation Systems for WMT 2017. In *Proceedings of the Second Conference on Machine Translation (WMT 2017), Volume 2: Shared Task Papers*, pages 374–381, Copenhagen, Denmark. Association for Computational Linguistics.

Mārcis Pinnis, Rihards Krišlauks, and Matīss Rikters. 2019. Tilde's Machine Translation Systems for WMT 2019. In *Proceedings of the Fourth Conference on Machine Translation*, pages 526–533, Florence, Italy. Association for Computational Linguistics.

Mārcis Pinnis, Matīss Rikters, and Rihards Krišlauks. 2018. Tilde's Machine Translation Systems for WMT 2018. In *Proceedings of the Third Conference on Machine Translation*, pages 477–485, Brussels, Belgium. Association for Computational Linguistics.

Roberts Rozis and Raivis Skadiņš. 2017. Tilde MODEL - Multilingual Open Data for EU Languages. In *Proceedings of the 21st Nordic Conference on Computational Linguistics*, pages 263–265.

Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán. 2019. Wikimatrix: Mining 135m parallel sentences in 1620 language pairs from wikipedia. *arXiv*, pages arXiv–1907.

Ralf Steinberger, Andreas Eisele, Szymon Klocek, Spyridon Pilos, and Patrick Schlüter. 2012. Dgt-tm: A freely available translation memory in 22 languages. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 454–459.

Ralf Steinberger, Bruno Pouliquen, Anna Widiger, Camelia Ignat, Tomaž Erjavec, Dan Tufi, and Dániel Varga. 2006. The JRC-Acquis: A Multilingual Aligned Parallel Corpus with 20+ Languages. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC'2006)*, volume 4, pages 2142—-2147.

Ole Tange. 2011. Gnu parallel - the command-line power tool. *The USENIX Magazine*, 36(1):42–47.

Jörg Tiedemann. 2016. Finding alternative translations in a large corpus of movie subtitle. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3518–3522.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Samsung R&D Institute Poland submission to WMT20 News Translation Task

Mateusz Krubiński, Marcin Chochowski, Bartłomiej Boczek,
Mikołaj Koszowski, Adam Dobrowolski,
Marcin Szymański, Paweł Przybysz
Samsung R&D Institute Poland

Abstract

This paper describes the submission to the WMT20 shared news translation task by Samsung R&D Institute Poland. We submitted systems for six language directions: English to Czech, Czech to English, English to Polish, Polish to English, English to Inuktitut and Inuktitut to English. For each, we trained a single-direction model. However, directions including English, Polish and Czech were derived from a common multilingual base, which was later fine-tuned on each particular direction. For all the translation directions, we used a similar training regime, with iterative training corpora improvement through back-translation and model ensembling. For the En $\rightarrow$ Cs direction, we additionally leveraged document-level information by re-ranking the beam output with a separate model.

1 Introduction

Since the Transformer architecture became the standard model in Neural Machine Translation, recent advancements in the field have come from two techniques. The first one is deepening the model by adding more layers, mainly in the encoder part, in order to model more complex dependencies (Raganato and Tiedemann, 2018; Wang et al., 2019a; Wu et al., 2019). This, however, poses problems during the training – too deep models are much harder to train due to the gradient vanishing problem (Zhang et al., 2019). The second technique consists in improving the quality of training data by removing spurious translations (Koehn et al., 2019) and making the data easier to learn through the teacher-student methodology (Hinton et al., 2015; Kim and Rush, 2016; Tan et al., 2019).

In this submission, we decided to leverage both techniques. We deepened the model with a lexical-shortcuts transformer modification. We also iteratively improved the synthetic corpora by training better and better translation models, back-translating and distilling the data in each step.

The remainder of this paper is structured as follows: Section 2 introduces the data used for training, Section 3 shows baseline NMT models and our experiments. In Section 4 we describe our training regime and results. Section 5 is for conclusions.

2 Data

2.1 Data Filtering

We used all the parallel data available in the constrained settings. We filtered the parallel data with a two-step process. First, we used a simple heuristics for general clean-up:

- remove pairs where any of the sentences is longer than 1500 characters
- remove sentences with characters not in the Unicode range specific to a given language pair
- remove pairs based on a length-ratio threshold.

We then de-duplicated the data and used the fast-align[1] tool to filter out pairs basing on the alignment probability between the source and the target (Table 1). For monolingual data, we used only the general clean-up procedure.

2.2 Data Pre-Processing

We used the `normalize-punctuation.perl`[2] script from the Moses package on all the training data. For the En $\leftrightarrow$ Iu directions, we used the alignment provided by the organizers, and

[1] github.com/clab/fast_align

[2] github.com/moses-smt/mosesdecoder/scripts/tokenizer/normalize-punctuation.perl

Proceedings of the 5th Conference on Machine Translation (WMT), pages 181–190

	Orig.	+ clean-up	+ fast-align
En ↔ Cs	62.5M	61.8M	43.4M
En ↔ Iu	2.6M	1.2M	1.1M
En ↔ Pl	11.2M	10.7M	8.6M

Table 1: Number of sentences in the parallel corpus originally, after simple rule-based cleaning-up, and after filtering out sentence pairs based on alignment probability.

decided to stick to the Inuktitut syllabics, without romanization.

For tokenization and segmentation, we used SentencePiece[3] (Kudo and Richardson, 2018). For the En ↔ Cs and En ↔ Pl directions, we started with a multilingual translation model that was later specialized towards each direction separately. For these 3 languages, we had to use a single, joint vocabulary with $32,000$ pieces and a unigram language model (ULM) tokenization scheme. For the En ↔ Iu directions, we used a joint vocabulary with the ULM tokenization scheme and $16,000$ pieces.

3 NMT System Overwiev

All of our systems are trained with the Marian NMT[4] (Junczys-Dowmunt et al., 2018) framework.

3.1 Baseline systems for En ↔ Cs and En ↔ Pl

We started with strong baselines, i.e. transformer models (Vaswani et al., 2017), which we will now refer to as *transformer-big*. This model consists of 6 encoder layers, 6 decoder layers, 16 heads, a model/embedding dimension of 1024 and a feed-forward layer dimension of 4096.

The model is regularized with a dropout between transformer layers of 0.2 and a label smoothing of 0.1. We also used layer normalization (Lei Ba et al., 2016) and tied the weights of the target-side embedding and the transpose of the output weight matrix, as well as source- and target-side embeddings (Press and Wolf, 2017). Optimizer delay was used to simulate bigger batches, updating weights every 16 batches, Adam (Kingma and Ba, 2015) was used as an optimizer, parametrized with a learning rate of 0.0003 and linear warm-up for the initial $32,000$ updates with subsequent inverted squared decay.

For each language pair, we trained both uni- and bi-directional models. We also examined the effect of using multilingual data to train a quadro-directional model on concatenated En ↔ Cs and En ↔ Pl corpora. The En ↔ Pl corpora were upsampled 5 times to match size. <*2XX*> tokens were appended to each sentence to indicate the target language. The results on newsdev2020 are presented in Table 2.

[3] github.com/google/sentencepiece
[4] github.com/marian-nmt/marian

	Uni	Bi	Quadro	Quadro-huge
En → Cs	26.1	25.4	24.4	26.0
Cs → En	32.4	31.4	30.5	32.7
En → Pl	26.1	25.4	26.2	27.3
Pl → En	30.0	30.3	31.0	32.3

Table 2: SacreBLEU scores on newsdev2020 for baseline trainings, for various model capacities: uni-directional models, bi-directional models and quadro-directional transformer-big. Quadro-huge stands for the quadro-directional model with the transformer-huge parameters.

	Corpora size	Pre-training	BLEU
En → Pl	0.25M	√	20.4
	0.25M	-	16.5
	0.5M	√	21.8
	0.5M	-	19.4
	8.6M	√	25.1
	8.6M	-	26.1
Pl → En	1M	√	27.1
	1M	-	25.7
	8.6M	√	29.1
	8.6M	-	30.0

Table 3: SacreBLEU scores on newsdev2020 for En ↔ Pl trainings, with and without pre-training. 8.6M corpora size means all the available training data was used.

3.2 Baseline system for En ↔ Iu

As the parallel corpora for En ↔ Iu are significantly smaller than for the other pairs, we decided to start with a transformer model with a smaller number of parameters i.e. *transformer-base*. All our base models were bi-directional.

The model consists of 6 encoder layers, 6 decoder layers, 8 heads, a model/embedding dimension of 512 and a feed-forward layer dimension of 2048. We examined the effect of vocabulary size on the model quality, and obtained the best results for the vocabulary size of $16,000$ (Table 4). Basing on our previous experience, we also examined an

	Vocab size	BLEU
En → Iu	16k	15.1
	32k	15.1
	64k	15.0
Iu → En	16k	28.3
	32k	27.9
	64k	27.6

Table 4: SacreBLEU scores on newsdev2020 for En ↔ Iu bi-directional trainings, for different sizes of the sentencepiece vocabulary.

unbalanced encoder/decoder configuration with a deeper encoder (8 layers) and a more shallow decoder (4 layers). The result was 28.3 (+0.0) for Iu → En and 15.3 (+0.2) for En → Iu, compared to the base case. We used this model as a reference for the following experiments.

3.3 Multilingual Denoising Pre-training

Liu et al. (2020) recently proposed a method for pre-training sequence-to-sequence models with an auto-encoder-based denoising objective. Pre-training a complete encoder-decoder model allows for later direct fine-tuning on the translation objective, with parallel corpora. In our experiment, we sampled 250M sentences from CommonCrawl for Czech, English and Polish (i.e. 750M in total). During training, we randomly cropped up to 25% tokens from each sentence, and taught the model to predict the original sequence. We used the same architecture as in baseline trainings. Next, we used the best checkpoint to warm-start training on the parallel data. Table 3 presents our results for varying sizes of the training corpus (the smaller corpus is a random subset of the parallel data). We observe that, although our implementation works well for low-resource setting, it leads to quality drop when all the parallel data is used. Accordlingly, we used this pre-training method only for the En ↔ Iu directions.

3.4 Lexical Shortcuts

Since our quadro-directional model showed promising results, we decided to try to examine the effect of deepening and enlarging the model. We increased the feed-forward layer dimension to 8192, and the number of encoder layers to 12. The rest of the parameters is the same as in *transformer-big*. He et al. (2019) demonstrated that, with a fixed number of layers, it was more efficient to have a deeper encoder than decoder. It also makes decoding for back-translation much faster. To help with gradient propagation, we implemented Lexical Shortcuts (Emelin et al., 2019) in the encoder. We used the feature-fusion version of the gating mechanism. The results are summarized in the Quadro-huge column in Table 2. This model outperformed the baseline in all the directions, except one. We decided to use this system in further trainings.

	Pl	En	Cs
Newscrawl	3.7M	230M	80.5M
+ Moore-Lewis	96.5M	-	-

Table 5: Number of sentences in monolingual datasets after clean-up and domain-based filtering.

3.5 Back-Translation with Language Model

Back-translation (Sennrich et al., 2016) is a common strategy of utilizing monolingual data in training NMT systems. For English and Czech, the amount of monolingual in-domain data in the Newscrawl data set is big enough, so for this language pair we used only the monolingual set. Yet for Polish, the Newscrawl is very limited in size, hence we decided to use Moore-Lewis filtering (Moore and Lewis, 2010) to extract in-domain data from CommonCrawl.

With this additional monolingual corpus, we had over 80M in-domain news sentences for each language (Table 5). We used those monolingual datasets to train an in-domain RNN-style language model for each of the three languages, using the same common vocabulary as the one in the translation models. This allowed us to easily ensemble this language model with a translation model during decoding, as described in Gulcehre et al. (2015). For each iteration of the back-translation, we used an ensemble of the top 4 NMT models available w.r.t. the dev-set score for the particular direction and the in-domain language model. The weights of the models were optimized through a grid-search.

3.6 Noisy Channel Model Reranking

Re-ranking the beam output is a method used to improve translation quality by the re-scoring hypothesis from a forward model. The noisy channel model (Yee et al., 2019) approach was used with success by Facebook in their submission to the WMT19 news translation task (Ng et al., 2019). Based on the Bayes' rule, given a target sequence y and a source sentence x, for every hypothesis from

the beam output, we calculate

$$\log P(y \mid x) + \lambda_1 \log P(x \mid y) + \lambda_2 \log P(y)$$

and use this score to re-rank the beam outputs. We model $P(y \mid x)$ with the forward model, $P(x \mid y)$ with the backward model and $P(y)$ with the language domain model. The weights λ_1 and λ_2 are tuned on the dev-set.

When we used this method for our baseline unidirectional systems, we noticed significant BLEU improvements: 26.9 (+0.8) on newsdev2020 for the En → Pl direction. However, there was no improvement when applied to translations produced with strong ensembles of both the domain language models and the translation models, trained on the back-translated data. In our final submission, we used this method only for the Iu → En and En → Iu directions.

3.7 Multi-Agent Dual Learning

In our submission, we used the simplified version of Multi-Agent Dual Learning (MADL) (Wang et al., 2019b), proposed in Kim et al. (2019), to generate additional training data from the parallel corpus. We generated n-best translations of both the source and the target sides of the parallel data, with strong ensembles of, respectively, the forward and the backward models. Next, we picked the best translation from among n candidates w.r.t. the sentence-level BLEU score. Thanks to these steps, we tripled the number of sentences by combining three types of datasets:

1. original source – original target,
2. original source – synthetic target,
3. synthetic source – original target,

where the synthetic target is the translation of the original source with the forward model, and the synthetic source is the translation of the original target with the backward model.

3.8 Document Level Reranking

For the En ↔ Cs translation directions, the training data is aligned on the document level. To make use of this information, we implemented the method presented in Voita et al. (2019). The method assumes one has access to consecutive tuples of sentences in the target language. Using the backward and forward models, one should translate the tuples with the sentence-level based systems, and then train the model to predict the original tuple, basing on the two-way translated data. As we already have access to the document-level aligned translations from the CzEng 2.0 corpus (Kocmi et al., 2020), we could do the translation just once. We experimented only with the En → Cs direction. We selected tuples of 4 consecutive sentences in English, translated each sentence independently, and glued the translations back together. We used a special token to indicate the end of the sentence. See Table 9 in the Appendix for an example of the training data. However, when we utilized this model to "repair" the newsdev2020 dev-set translations, we noticed a quality drop. We decided to try a different approach, and used the document-level repair model to re-rank the beam output. The procedure is similar to a greedy search for the best path through n-best lists of forward model translations. It is described with Algorithm 1.

Algorithm 1 Document Level Reranking

Input: $\{trn^b(s_j)\}$ - n-best list ($b = 1..N$) with translations of sentence s_j

Input: $L_{repair}(\{sa_j\}_{j=1..4}, \{sb_j\}_{j=1..4})$− likelihood computed with repair model for two 4-sentence sequences

Output: Re-ranked translations rep

for all paragraph in test-set **do**
  $i = 0$
  for all sentence s_i in paragraph **do**
    if $i < 4$ **then**
      $rep_i = trn^1(s_i)$
    else
      $seq_1 = rep_{i-3}, \ldots, rep_{i-1}, trn^1(s_i)$
      $seq_b = rep_{i-3}, \ldots, rep_{i-1}, trn^b(s_i)$
      $rep_i = \arg\max_{b=1..N} L_{repair}(seq_1, seq_b)$
    end if
    $i \mathrel{+}= 1$
  end for
end for

Although on the dev-set we didn't see much difference in the BLEU score, manual inspection showed some promising results. We decided to apply this method to our best-scoring system and saw a 0.1 improvement in the BLEU score on the test-set.

3.9 Post-Processing

For all the translation directions we participated in, we normalized the system outputs with a series of

regular expressions:

- substitute English quotation marks (“ ... ”) with Czech/Polish ones („ ... ”),
- if a source starts/ends with a quotation mark, we make sure so does the translation,
- remove word repetitions,
- replace consecutive sequences of white-spaces with a single one,
- if a source ends with a punctuation mark (e.g. ?.!), we substitute the last character of the translation with it,
- replace three consecutive dots with an ellipsis,
- replace hyphens with en dashes.

4 Results

4.1 English → Polish

The model for the English → Polish direction was derived from the multilingual quadro-huge model – similarly to the other models for directions with Polish, Czech or English. The successive steps and respective BLEU scores are reported in Table 6.

We started with fine-tuning the quadro-directional model on the parallel data for the specific direction. Next, we used an ensemble of our best models to back-translate Newscrawl 2018 and 2019, we filtered it (3.5M sentences) and merged with the parallel corpus (8.6M). The fine-tuning gave us +1.5 BLEU improvement. We were able to achieve an additional +0.9 BLEU with the rule-based post-processing (see above). In the next step, we used the MADL procedure to generate additional data. To further increase the amount of data and its variability, we picked the top 2 best translations, according to the sentence-level BLEU in the distillation process – instead of choosing just one. Again, we up-sampled the original parallel corpus twice. This procedure gave additional 52M sentences (a 6-fold increase).

We back-translated all the monolingual in-domain data (i.e. 89M after filtration) and used both corpora to fine-tune the next generation model. We augmented the data by randomly masking up to 10% of the input tokens with a random punctuation mark, and observed yet another performance boost. Using all these corpora, we trained another model from scratch, hoping to get a less correlated model.

	newsdev2020	
System	**En → Pl**	**Pl → En**
Quadro-huge	27.3	32.3
+ finetune	27.4	32.8
+ ensemble	28.7	32.9
+ BT	28.9	33.7
+ post-process	29.8	-
+ ensemble	30.7	34.1
+ BT2 & MADL	31.4	34.4
+ masking	31.6	-
FRESH	30.2	32.9
+ ensemble	32.2	34.9
+ post-process	-	35.0
+ test-dev tune	32.2	35.1
+ ensemble	32.4	35.4
	newstest2020	
WMT'20 SUBMISSION	**27.6**	**34.3**

Table 6: Successive improvements in the BLEU scores on the English → Polish and Polish → English directions, computed with SacreBLEU.

Although the fresh model performance was poorer than the previous best (30.2 BLEU vs. 31.6 BLEU), the grid-search ensemble optimization included it in the best ensemble. As a final step, we used Moore-Lewis filtering to choose 1M Newscrawl sentences that were closest to the concatenated newsdev2020 and newstest2020. We translated them with the best ensemble, and used it to fine-tune our best-performing model. Again, we ran ensemble optimization including this model into the models reservoir. The optimal ensemble was the one we submitted as the primary system.

4.2 Polish → English

For the Polish to English direction, we proceeded similarly to our solution for English to Polish. We started with the quadro-huge model. We back-translated Newscrawl 2018, filtered it (12M sentences) and merged with the parallel corpus (8.6M). We kept on training the fine-tuned quadro-huge model, increasing the performance by 0.9 BLEU. We used the same MADL procedure as before, distilling 2 best translations for each source sentence. We also back-translated Newscrawl 2007-2017 (144M) and merged it with the MADL corpus. With this corpus, single model performance increased by +0.7 BLEU. We used the same corpus to train a fresh model. Similarly to the English to Polish direction, the fresh model performed poorer

(32.9 BLEU) than the fine-tuned one (34.4 BLEU), but – again – in ensemble it gave additional improvement. Finally, we fine-tuned on the 1M corpora filtered out from Newscrawl in the domain of the concatenated newsdev2020 and newstest2020, and ensembled for the final submission.

4.3 English → Czech

We started with fine-tuning the quadro-huge model with only English to Czech parallel data and ensembling several models into one. This model specialization gave us +1.4 BLEU. Next, we back-translated Newscrawl 2018 and 2019 in two flavors: normally, and with adding Gumbel noise (`--output-sampling` in Marian). Then, we filtered the result (see section 2.1), obtaining 35M sentences. With this additional corpus, the single model performance improved by 0.9 BLEU. In contrast to the Cs → En direction, using back-translations from CzEng 2.0 seemed to hurt the model performance.

In the next iteration, we produced the MADL corpus (120M) and merged it with back-translated Newscrawl 2009-2017 (102M, with and without noise) and used this data to train yet another model. Finally, we ensembled this model with models trained from scratch and fine-tuned on the 1M from Newscrawl common with the concatenated newsdev2020 and newstest2020. Before the last step – document level re-ranking – we used the sentence-splitter from NLTK (Bird et al., 2009) to pre-process the testset. It was required because of our systems being trained with sentence-level data and in newstest2020 some of the segments contain multiple sentences. We translated the pre-processed testset with the best ensemble, re-ranked on the document level and finally glued back the translations together. The document level re-ranking gave us -0.1 BLEU on the dev-set but +0.1 on the test-set.

4.4 Czech → English

Again, the specialization of the quadro-huge model with only Czech to English data gave us almost 1 BLEU gain in performance. Next, we back-translated Newscrawl 2018 and 2019 and filtered it with our pipeline, obtaining 49M sentences. We added the Newscrawl translations from CzEng 2.0 (79M) and the original filtered parallel corpus (43M), ending up with 171M parallel sentences as our training set. Using this data, we improved the single model performance by 3.6 BLEU. Fine-tuning on the MADL corpus (120M) and the back-translated Newscrawl 2017 (25M) gave additional +1.2 BLEU on the single model. Finally, we ensembled them with a model trained from scratch and fine-tuned on the 1M sentences from Newscrawl that were similar to the concatenated newsdev2020 and newstest2020 w.r.t. the Moore-Lewis score. For sentence splitting, we used the same splitter as for En → Cs. Results on the previous test-sets of the final systems for the En ↔ Cs directions, without document level re-ranking, in the Appendix (Table 10).

	newsdev2020	
System	**En → Cs**	**Cs → En**
Quadro-huge	26.0	32.7
+ finetune	26.5	33.5
+ ensemble	27.3	33.8
+BT	27.4	37.4
+ ensemble	28.5	37.7
+ post-process	-	37.8
+ BT2 & MADL	28.8	38.6
FRESH	27.0	35.6
+ ensemble	29.1	38.7
+ test-dev tune	29.1	39.0
+ ensemble	29.4	39.7
+ post-process	31.3	-
+ doc-level re-rank	31.2	-
	newstest2020	
WMT'20 SUBMISSION	**36.5**	**28.5**

Table 7: Successive improvements in the BLEU scores on the English → Czech and Czech → English directions, computed with SacreBLEU.

4.5 English ↔ Inuktitut

In contrast to all the other directions, for Inuktitut we had much less monolingual data (10k after cleaning) than bitext (1.1M). In the first step, we back-translated the monolingual data with beam 10, and kept all the possible variants (0.1M sentences). We also back-translated the English Europarl v10 corpus (2M), because we believed it to help with the Hansard (Joanis et al., 2020) part of the dev- and test-sets. We merged it with the two-directional parallel data (2.2M) and trained a bi-directional model, from scratch. We used it for the general first iteration of the MADL corpus (6.6M), and used all of the data to once more train a model from scratch. Here we examined the effect of pre-training. We used the sample (10M)

	newsdev2020	
System	**En → Iu**	**En → Iu**
Baseline	15.3	28.3
+ BT	15.5	29.9
+ MADL	15.5	30.4
+ masked mono	15.8	30.5
+ transformer-big	15.8	32.2
+ fine-tune	15.8	32.5
+ ensemble	16.2	32.7
+ MADL2	15.8	32.7
+ ensemble	16.3	32.9
+ noisy channel	16.4	33.2
	newstest2020	
WMT'20 SUBMISSION	**11.0**	**25.6**

Table 8: Successive improvements in the BLEU scores on the English → Inuktitut and Inuktitut → English directions, computed with SacreBLEU.

from the English Newscrawl 2019 and the Inuktitut part of the parallel data, up-sampled 5 times (5.5M). With the pipeline approach, fine-tuning on bitext was giving us similar results as training on bitext from scratch. Nevertheless, we were however able to achieve some improvement, when training a fresh model on the merged parallel and noised monolingual data. We were able to achieve further improvement with increased model size – 1024 embedding dimension, 4096 forward dimension and 16 heads.

Next, we started fine-tuning on each direction independently, using the parallel data for En → Iu, and 20-times up-sampled the parallel data (22M) together with the back-translated Newscrawl 2018 and 2019 (48M) for Iu → En. Then, we used an ensemble of models to once again generate the MADL corpus, use it to fine-tune the unidirectional models and the ensemble once again. We used the Noisy Channel Reranking method and saw some improvement on both the dev-set and the test-set.

5 Conclusions

In this paper, we have described the submission to the WMT20 shared news translation task by Samsung R&D Institute Poland. All submitted systems were constrained and utilized only the permitted data. With our approach, we were able to leverage two important techniques that improve the translation quality. One method was deepening the model, while still being able to train it effectively. The other one was filtering and improving the quality of the training data and producing high quality synthetic data. Our iterative approach of improving the training data and improving the translation model proved to be successful, showing gradual increase in the BLEU scores.

References

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python*, 1st edition. O'Reilly Media, Inc.

Denis Emelin, Ivan Titov, and Rico Sennrich. 2019. Widening the Representation Bottleneck in Neural Machine Translation with Lexical Shortcuts. In *Proceedings of the Fourth Conference on Machine Translation*, pages 102–115, Florence, Italy. Association for Computational Linguistics.

Caglar Gulcehre, Orhan Firat, Kelvin Xu, Kyunghyun Cho, Loic Barrault, Huei-Chi Lin, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2015. On Using Monolingual Corpora in Neural Machine Translation. *arXiv e-prints*, page arXiv:1503.03535.

Tianyu He, Xu Tan, and Tao Qin. 2019. Hard but Robust, Easy but Sensitive: How Encoder and Decoder Perform in Neural Machine Translation. *arXiv e-prints*, page arXiv:1908.06259.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the Knowledge in a Neural Network. *arXiv e-prints*, page arXiv:1503.02531.

Eric Joanis, Rebecca Knowles, Roland Kuhn, Samuel Larkin, Patrick Littell, Chi-kiu Lo, Darlene Stewart, and Jeffrey Micher. 2020. The Nunavut hansard Inuktitut–English parallel corpus 3.0 with preliminary machine translation results. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 2562–2572, Marseille, France. European Language Resources Association.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia. Association for Computational Linguistics.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327, Austin, Texas. Association for Computational Linguistics.

Young Jin Kim, Marcin Junczys-Dowmunt, Hany Hassan, Alham Fikri Aji, Kenneth Heafield, Roman Grundkiewicz, and Nikolay Bogoychev. 2019. From

research to production and back: Ludicrously fast neural machine translation. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 280–288, Hong Kong. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *The International Conference on Learning Representations*, San Diego, California, USA.

Tom Kocmi, Martin Popel, and Ondrej Bojar. 2020. Announcing czeng 2.0 parallel corpus with over 2 gigawords. *arXiv preprint arXiv:2007.03006*.

Philipp Koehn, Francisco Guzmán, Vishrav Chaudhary, and Juan Pino. 2019. Findings of the WMT 2019 shared task on parallel corpus filtering for low-resource conditions. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 54–72, Florence, Italy. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer Normalization. *arXiv e-prints*, page arXiv:1607.06450.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual Denoising Pre-training for Neural Machine Translation. *arXiv e-prints*, page arXiv:2001.08210.

Robert C. Moore and William Lewis. 2010. Intelligent selection of language model training data. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 220–224, Uppsala, Sweden. Association for Computational Linguistics.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook FAIR's WMT19 news translation task submission. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 314–319, Florence, Italy. Association for Computational Linguistics.

Ofir Press and Lior Wolf. 2017. Using the output embedding to improve language models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 157–163. Association for Computational Linguistics.

Alessandro Raganato and Jörg Tiedemann. 2018. An analysis of encoder representations in transformer-based machine translation. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 287–297, Brussels, Belgium. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Xu Tan, Yi Ren, Di He, Tao Qin, and Tie-Yan Liu. 2019. Multilingual neural machine translation with knowledge distillation. In *International Conference on Learning Representations*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Elena Voita, Rico Sennrich, and Ivan Titov. 2019. Context-aware monolingual repair for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 877–886, Hong Kong, China. Association for Computational Linguistics.

Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F. Wong, and Lidia S. Chao. 2019a. Learning deep transformer models for machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1810–1822, Florence, Italy. Association for Computational Linguistics.

Yiren Wang, Yingce Xia, Tianyu He, Fei Tian, Tao Qin, ChengXiang Zhai, and Tie-Yan Liu. 2019b. Multi-agent dual learning. In *International Conference on Learning Representations*.

Lijun Wu, Yiren Wang, Yingce Xia, Fei Tian, Fei Gao, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. 2019. Depth growing for neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5558–5563, Florence, Italy. Association for Computational Linguistics.

Kyra Yee, Yann Dauphin, and Michael Auli. 2019. Simple and effective noisy channel modeling for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5696–5701, Hong Kong, China. Association for Computational Linguistics.

Biao Zhang, Ivan Titov, and Rico Sennrich. 2019. Improving deep transformer with depth-scaled initialization and merged attention. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 898–909, Hong Kong, China. Association for Computational Linguistics.

A Appendix

source	”To tys vymyslel mihotavé reflektory?” <SEP> ”...Ne. <SEP> V nanouffu nejsem moc dobrá. <SEP> Přišel na ně můj přítel z Londýna.
target	”A vymyslel jste taky ty reflektorky?” <SEP> ”Ne. <SEP> V nanotechnologii tak dobrý nejsem. <SEP> S tím přišel jeden můj známý z Londýna.
source	A na čest svého domu, prohlašuji, že můj milovaný Robert.. . . -Určitě? <SEP> Radši se podepiš s Jaimem Lannisterem, Králokatem. <SEP> To město je nudný. <SEP> Prosím, Andrewe.
target	”A já prohlašuji na čest svého rodu, že můj milovaný bratr Robert...” <SEP> Dej tam serem Jaimem Lannisterem, Králokatem. <SEP> Tohle město páchne... <SEP> Prosím!
source	Řekla jsem: ”Čí byl nápad?” <SEP> Jejich modré oči byly jasné jako plavecký bazén. <SEP> ”To přišel točení Ernesto.” vzdálený Dezertér nebo lhář. <SEP> Byli jste dost dobří přátelé?”
target	”Čí to byl nápad?” zeptala jsem se. <SEP> Podíval se na mě zpříma a jeho modré oči byly průzračné jak studánky. <SEP> ”Earnesto s tím přišel.” <SEP> ”Byli jste dobří kamarádi?”

Table 9: Example of the training data used to train the document-level re-rank model. Target is a quadruple of consecutive sentences extracted from the CzEng 2.0 parallel corpus. Source is a translation of the matching English sequence, produced on the sentence level.

		En → Cs	**Cs → En**
newstest2019	WMT‘19 best	29.9	-
	SRPOL‘20	31.3 (+1.4)	-
newstest2018	WMT‘18 best	26.0	33.9
	SRPOL‘20	27. 4 (+1.4)	35.3 (+1.4)
newstest2017	WMT‘17 best	26.1	30.9
	SRPOL‘20	27.7 (+1.6)	35.1 (+4.2)

Table 10: SacreBLEU scores of the final systems for the En ↔ Cs directions, without document level re-ranking, on test-sets from previous years.

Speed-optimized, Compact Student Models that Distill Knowledge from a Larger Teacher Model: the UEDIN-CUNI Submission to the WMT 2020 News Translation Task

Ulrich Germann[⋆], **Roman Grundkiewicz**[⋆], **Martin Popel**[‡], **Radina Dobreva**[⋆], **Nikolay Bogoychev**[⋆], **Kenneth Heafield**[⋆]
[⋆] University of Edinburgh, UK
[‡] Charles University, Prague, Czech Republic
{ugermann|rgrundki|rdobreva|nbogoych|kheafiel}@inf.ed.ac.uk
popel@ufal.mff.cuni.cz

Abstract

We describe the joint submission of the University of Edinburgh and Charles University, Prague, to the Czech/English track in the WMT 2020 Shared Task on News Translation. Our fast and compact student models distill knowledge from a larger, slower teacher. They are designed to offer a good trade-off between translation quality and inference efficiency. On the WMT 2020 Czech ↔ English test sets, they achieve translation speeds of over 700 whitespace-delimited source words per second[1] on a single CPU thread, thus making neural translation feasible on consumer hardware without a GPU.

1 Introduction

The conventional set-up of the WMT Shared Tasks on News Translation emphasizes translation quality (however measured) above all else. Constraints on the data that may be used for training in the 'constrained' track establish a level playing field in terms of the information available to the translation model and its training process, but there are no constraints on the computational power and effort spent to achieve the results. In contrast, the WNGT Shared Task on Efficient Translation (Heafield et al., 2020) encourages participants to submit systems that are both accurate and efficient during inference (i.e., translation). So far, there has been little interaction between the two tasks.

With our joint submission between the University of Edinburgh (UEDIN) and Charles University, Prague (CUNI), we strive to bridge this gap. We submitted small, efficient systems that distilled knowledge from a more powerful teacher model via sequence-level knowledge distillation (Kim and Rush, 2016). In a nutshell, the procedure can be described as follows:

1. Train a powerful teacher model on the available training data set D.
2. Translate the source side of D plus available monolingual data in the source language and appropriate text domain with the teacher model to generate the training set D'.
3. Train a small student model on D'.

While the computational effort to first train a teacher model and then distill its knowledge into a student model is considerably greater than just training the teacher — in addition to training the teacher, we have to translate the training data and then train a student on that data —, the advantage is at inference time. Even the larger of the two student models we present in this paper can translate on a single CPU core at an acceptable speed (cf. Tab. 4). Translation can further be sped up by quantizing parameters to 8 bits of precision and using Integer General Matrix Multiplication (IntGEMM) for inference. Even though our submissions to the WMT 2020 Shared Task on News Translation were produced by unquantized floating point models, we report performance numbers for quantized models as well to demonstrate their efficacy and show that they can speed up translation by about 10% with negligible trade-offs in terms of BLEU score over unquantized models.

All models used in this work are based on the Transformer architecture (Vaswani et al., 2017). Details are discussed in the sections below; the hyper-parameter settings for each model are listed in Tab. 1. The student models described in this paper can be obtained via https://github.com/browsermt/students.

2 Teacher Models

The teacher models were produced by CUNI, using the Tensor2Tensor deep-learning toolkit (Vaswani et al., 2018)[2]. The teachers were trained on the full

[1] Bogoychev et al. (2020) report translation speeds of up to 3135 source words per second on a single CPU thread; the actual throughput depends not only on the computer hardware used for translation but also on the distribution of translation segment lengths in the test set.

[2] https://github.com/tensorflow/tensor2tensor

Proceedings of the 5th Conference on Machine Translation (WMT), pages 191–196
Online, November 19–20, 2020.

Table 1: Transformer hyper-parameters for T2T teacher and Marian student models.

	teacher		student	
parameter	**cs→en**	**en→cs**	**base**	**tiny**
vocabulary size (spm)	32K	32K	32K	32K
joint vocabulary	yes	yes	yes	yes
encoder layers	6	12	6	6
decoder layers	6	6	2	2
decoder auto-reg.	self-attention	self-attention	SSRU	SSRU
tied embeddings	yes	yes	yes	yes
embedding size	1024	1024	512	256
filter size	4096	4096	2048	1536
number of att. heads	16	16	8	8
att. key size	64	64	64	64
att. value size	64	64	64	64
checkpoints avg.	8	8	exp. smoothing[a]	
back-translation	block-BT	block BT	none	none
beam search alpha	1.0	1.0	1.0	1.0
max training length	150	150	200	200

[a] Exponential smoothing with $\alpha = 0.0001$.

CzEng 2.0 dataset (Kocmi et al., 2020)[3] consisting of genuine (authentic) parallel data as well as monolingual news data translated by CUNI's transformer systems from WMT 2018 (Popel, 2018) to generate back-translated synthetic training data (Sennrich et al., 2016). Rather than shuffling and mixing authentic and synthetic training data, the teacher models were trained on alternating blocks of authentic and synthetic data ("block-regime back-translation" (block-BT); Popel et al., 2020), spending about 10 hours of training time on each block.

The model parameters for the final teacher models were obtained by checkpoint averaging over the last 8 checkpoints of the training process, saved in hourly intervals.

The en→cs teacher model used in this work also produced CUNI's primary submission to the WMT 2020 Shared Task on News Translation ("CUNI-Transformer"; Popel, 2020). However, the CUNI submission used a beam size of 4 instead of 8 as used in this work, resulting in a BLEU score on the WMT 2020 en→ test set that is 0.2 lower than the BLEU score reported in Tab. 4.

The cs→en teacher model used in this work has only 6 encoder layers as opposed to the 12 encoder layers used in CUNI's primary submission to the Shared Task, resulting in a BLEU score on the WMT 2020 test set that is 1.0 BLEU points lower than the score achieved by the model used for CUNI's primary submission.

3 Student Models

The smaller, more efficient student models were trained by UEDIN with the Marian NMT toolkit (Junczys-Dowmunt et al., 2018a)[4]. The students were trained on artificial training data produced by knowledge distillation (Kim and Rush, 2016), where the target side of the parallel data is the teacher model's translation of the source side. The basic idea is that the teacher guides the student towards translations that can be achieved with the teacher's knowledge.

3.1 Student Model Architectures

The student models use the architecture proposed by Kim et al. (2019) with improvements by Bogoychev et al. (2020). Apart from using fewer layers and fewer dimensions in each layer, the main difference of the students from the conventional transformer architecture is the use of Simpler Simple Recurrent Units (SSRU; Kim et al., 2019) instead of the self-attention mechanism in decoder part of the transformer. For the sake of simplicity, our student models use exponential smoothing of model parameters with a smoothing parameter of 0.0001 instead of the checkpoint averaging used to produce the final teacher models.

For each translation direction, we trained two models: a base transformer and a 'tiny' transformer with fewer decoder layers and a smaller number of embedding and filter dimensions; specifications are shown in Tab. 1.

3.2 Data Preparation

To create artificial training data for the students, we used the original parallel section of the CzEng 2.0 dataset but no back-translations. Instead, we translated ca. 40 million sentences from the mono-

[3] http://ufal.mff.cuni.cz/czeng

[4] https://github.com/marian-nmt/marian

Table 2: Data used for training the models (in millions of sentence pairs).

data set	teacher		student	
	cs→en	en→cs	cs→en	en→cs
CzEng 2.0 parallel (original)	61.1m	61.1m		
CzEng 2.0 parallel (pre-filtered)			42.3m	42.3m
Back-translated news (CzEng 2.0 'mono')	50.6m	76.2m		
Teacher-translated news			50.1m	43.0m
Top 90% according to alignment score			83.2m	76.8m
Total used	111.7m	137.3m	83.2m	76.8m

lingual English NewsCrawl corpus[5] (2018 and 2019) for en→cs and 50 million sentences from the monolingual Czech NewsCrawl corpus (2013–2019) for cs→en.

Prior to translation with the respective teacher model, we filtered and de-duplicated the data. Filtering consisted of the following steps:

- Sentence-level deduplication.
- Removal of excessively long sentences (longer than 120 space-separated tokens; note that the sentence length limit for training in terms of subword units was 200; cf Tab. 1).
- Removal of sentence pairs that were not identified as the correct language by the fastText language identifier (Joulin et al., 2017, 2016) Python module.[6]
- For parallel data, removal of sentence pairs with length ratio larger than 2.5 (in terms of words of untokenized text), i.e. the longer sentence could not be more than 2.5 times as long as the shorter one.
- Removal of sentences in which less than half the words contain an alphabetical character or less than half the characters belong to the alphabet of the specific language.

We translated the cleaned data with the Tensor2Tensor teacher model with a beam size of 8. A small proportion of 'odd' sentences that had escaped our cleaning process, for example sentences with several long URLs that resulted in very long token sequences after segmentation into subword units, forced us to use a relatively small batch size of 8–24 sentences to avoid out-of-memory errors. For load balancing, we split the translation load into blocks of 10,000 sentences each and parallelized the translation process over dozens of machines. Using comparatively many translation blocks gave us flexibility in scaling the translation operation in response to resource availability.

[5] http://data.statmt.org/news-crawl/
[6] https://pypi.org/project/fasttext/

Despite the small batch size, 32 of our 10,0000-sentence input chunks still failed to translate due to memory limitations. A cursory investigation revealed that these often contained undesirable material (such as Javascript and HTML code that had somehow survived the filtering process), so that we decided to simply discard those blocks of data.

We made no effort to optimize translation speed and throughput for the teacher models in Tensor2Tensor; translation time for a single 10,000-sentence block was ca. 30–45 minutes, depending on sentence lengths and hardware used.

Table 3: Distribution of teacher-produced translations chosen by sentence-level BLEU score over their respective ranks in the decoder beam.

rank	en→cs	cs→en
1	32.22%	31.24%
2	15.20%	15.63%
3	12.25%	12.21%
4	9.79%	9.87%
5	8.89%	8.88%
6	7.73%	7.85%
7	7.26%	7.40%
8	6.67%	6.93%

For the authentic parallel data, we selected from the 8 top-scoring final translation hypotheses for each source sentence the one with the highest sentence-level BLEU score with respect to the original target side of the data. Table 3 shows the distribution of the hypotheses selected over the respective beam ranks. For the monolingual data, for which we obviously have no human reference translations, we simply chose the highest-scoring translation. Sentence pairs where the translation contained the same whitespace-separated sequence of words three or more times in a row, or the same sequence of one or more characters in five or more subsequent repetitions (which can happen when the recursive decoder goes into a loop) were discarded.

We subsequently tokenized the synthetic teaching data (source and translations by the teacher model) with SentencePiece (Kudo and Richardson, 2018),

using a joint vocabulary for both languages with a size of 32,000 tokens. This vocabulary is also used by the final systems. The tokenized training data was word-aligned in both translation directions with FastAlign (Dyer et al., 2013). Directional word alignments were then symmetrized with the grow-diag-final-and symmetrization algorithm (Koehn et al., 2003).

These word alignments serve mainly three purposes: (a) to guide the attention mechanism during training of the student models (Liu et al., 2016) with guided alignment (Chen et al., 2016); (b) to produce shortlists of translation candidates to limit the choice of target words that need to be considered during inference (Junczys-Dowmunt et al., 2018b); and (c) to give us a rough estimate of translation quality via average per-token alignment scores for each sentence pair. We used these scores to discard the bottom 10% of our artificial training data.

Based on our experiments, the guided alignment training is neither required for student model training, nor does it improve BLEU scores on the development set. However, it encourages the guided decoder layer to mimic word alignments, which can be useful in post-processing.[7] We used default settings from Marian for the guided alignment training.

3.3 Quantized Models

Floating point operations are computationally more expensive than integer operations. However, as Han et al. (2016) have shown, neural network inference does not require the high precision of representation and computation that 32-bit floating point numbers offer. Devlin (2017) suggests a simple quantization mechanism for quantizing parameters to 16-bit integer precision and notes that support for off-the-shelf 8-bit integer matrix multiplication is lacking. Bogoychev et al. (2020) fill that gap and provide an 8-bit quantization and fine tuning scheme for Marian based on the intgemm library;[8] we used that scheme for our models. The model parameters are quantized offline from *float32* to *int8*, and during translation, the activations are quantized just prior to each GEMM operation. The GEMM operation is performed in 8-bit integers, and then the result is de-quantized back to *float32*. Despite the extra quantization and de-quantization involved, the increased speed at which 8-bit integer multiplication is performed more than compensates for it. Bogoychev et al. (2020) observe that smaller student presets lose BLEU when quantized. In order to counteract that, we perform model fine tuning following the work of Aji and Heafield (2020): We replace the GEMM routine implementation with a custom one that is damaged, according to the quantization scheme and perform several thousand mini-batch updates of the model. The damaged GEMM implementation can only produce 255 unique float values (corresponding to the 8-bit integer dequantization range) and the model quickly learns to work with those values and recovers some of the BLEU lost compared to untuned quantized model.

4 Results

In Table 4, we show the performance of the three models in terms of BLEU scores for the WMT 2020 cs↔en test sets and translation speed. Teacher models ran on an Nvidia GeForce GTX 1080 with a batch size of 16. Student models were run on a single CPU core on an Intel Intel(R) Xeon(R) CPU E5-2680 0 @ 2.70GHz with a batch size of 64. It should be noted that we made no effort to optimize the teachers for translation speed.

Text segments in the WMT 2020 cs↔en test sets are aligned at the paragraph level; we therefore split the provided segments into individual sentences prior to translation with Moses-style sentence splitting[9] and restored paragraphs afterwards.

All BLEU scores were computed with SacreBLEU.[10] For the en→cs teacher model, removing repetitions and adapting quotation marks to Czech spelling conventions boosted the BLEU score by 1.6 BLEU points; for student models, this post-processing is not necessary. Having been trained on post-processed teacher output, the student models have learned this correctly. Except where indicated, translation was with a greedy search (beam size 1) and a shortlist of 50 translation candidates per source word.

Due to resource congestion, we were not able to fully train the models by the submission deadline; our submissions are based on the systems with the best validation BLEU score available at the time. For validation, we used the concatenation of all parallel data for the respective translation direction from the WMT test sets from the years 2008 through 2019 where the original language of the data is the source language for the translation direction in question.

In terms of BLEU score on the WMT 2020 test set, the submitted primary system for cs→en is ca. 0.5 BLEU points below the final system; the en→cs system incidentally outperforms the final system by 0.5 BLEU points, as shown in Tab. 4.

[7] For example, for handling HTML tags in translated texts.

[8] https://github.com/kpu/intgemm

[9] https://github.com/ugermann/ssplit-cpp, which is a C++ reimplentation of the Moses sentence splitter, currently covering only a subset of the languages supported by the Moses Sentence splitter (no non-roman scripts).

[10] Post (2018); BLEU+case.mixed+lang.${src}-${trg}+numrefs.1 +smooth.exp+test.wmt20+tok.13a +version.1.4.13

Table 4: BLEU results for teacher and student models (base, tiny) on the WMT20 test set.

system	cs→en BLEU	cs→en time[a]	cs→en words/sec.	en→cs BLEU	en→cs time	en→cs words/sec.
teacher (no postprocessing)[b]	27.6	782 sec.	33	34.0	1131 sec.	39
teacher (with postprocessing)[b]				35.6	1131 sec.	39
base (float32, primary sub.)[b]	27.7[c]	424 sec.	61	36.3[d]	637 sec.	69
base (float32)[b]	28.2	294 sec.	88	35.8	465 sec.	95
base (float32)	27.9	101 sec.	256	35.7	151 sec.	292
base (8-bit quant., untuned)	27.5	90 sec.	287	34.4	136 sec.	324
base (8-bit quant., tuned)	27.8	88 sec.	294	35.3	136 sec.	324
base (8-bit quant., tuned, precomp[e])	27.9	89 sec.	291	35.7	135 sec.	326
tiny (float32)	27.0	38 sec.	681	34.7	59 sec.	746
tiny (8-bit quant., untuned)	25.6	34 sec.	761	31.9	55 sec.	815
tiny (8-bit quant., tuned)	26.9	35 sec.	739	32.9	55 sec.	815
tiny (8-bit quant., tuned, precomp[e])	26.9	35 sec.	739	32.8	53 sec.	830

[a] Inference time for core test set without additional test sets.
[b] Beam size 8; postprocessing: remove repetitions, adapt quotation marks to Czech conventions.
[c] After 65K updates, shortlist size 100.
[d] After 190K updates, shortlist size 100.
[e] Pre-computed scaling factor for quantization, see Sec. 5.1 in Bogoychev et al. (2020) for details.

5 Conclusion

We presented student models that distill knowledge from a larger teacher model without loss in BLEU performance. (In fact, for the WMT 2020 test set, our larger student models technically outperform the teacher in terms of BLEU, but we consider that difference accidental.) At the same time, they are significantly faster and do not require a GPU for inference, making it possible to perform neural machine translation on consumer-grade hardware without the use of a GPU.

Acknowledgements

This work was supported by funding from the European Union's Horizon 2020 research and innovation programme under grant agreements No 825303 (Bergamot) and 825627 (European Language Grid).

This work was performed using resources provided by the Cambridge Service for Data Driven Discovery (CSD3) operated by the University of Cambridge Research Computing Service (http://www.csd3.cam.ac.uk/), provided by Dell EMC and Intel using Tier-2 funding from the Engineering and Physical Sciences Research Council (capital grant EP/P020259/1), and DiRAC funding from the Science and Technology Facilities Council (www.dirac.ac.uk). The work has been using language resources developed and distributed by the LINDAT/CLARIAH-CZ project of the Ministry of Education, Youth and Sports of the Czech Republic (project LM2018101).

References

Aji, Alham Fikri and Kenneth Heafield. 2020. Compressing neural machine translation models with 4-bit precision. In *Proceedings of the Fourth Workshop on Neural Generation and Translation*, pages 35–42, Online. Association for Computational Linguistics.

Bogoychev, Nikolay, Roman Grundkiewicz, Alham Fikri Aji, Maximiliana Behnke, Kenneth Heafield, Sidharth Kashyap, Emmanouil-Ioannis Farsarakis, and Mateusz Chudyk. 2020. Edinburgh's submissions to the 2020 Machine Translation Efficiency Task. In *Proceedings of the Fourth Workshop on Neural Generation and Translation*, pages 218–224, Online. Association for Computational Linguistics.

Chen, Wenhu, Evgeny Matusov, Shahram Khadivi, and Jan-Thorsten Peter. 2016. Guided alignment training for topic-aware neural machine translation. *Proceedings of AMTA 2016*.

Devlin, Jacob. 2017. Sharp models on dull hardware: Fast and accurate neural machine translation decoding on the CPU. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2820–2825, Copenhagen, Denmark. Association for Computational Linguistics.

Dyer, Chris, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648, Atlanta, Georgia. Association for Computational Linguistics. Code at `https://github.com/clab/fast_align`.

Han, Song, Huizi Mao, and William J. Dally. 2016. Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Heafield, Kenneth, Hiroaki Hayashi, Yusuke Oda, Ioannis Konstas, Andrew Finch, Graham Neubig, Xian Li,

and Alexandra Birch. 2020. Findings of the fourth workshop on neural generation and translation. In *Proceedings of the Fourth Workshop on Neural Generation and Translation*, pages 1–9, Online. Association for Computational Linguistics.

Joulin, Armand, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016. Fasttext.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

Joulin, Armand, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431, Valencia, Spain. Association for Computational Linguistics.

Junczys-Dowmunt, Marcin, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018a. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia. Association for Computational Linguistics. Code at `https://github.com/marian-nmt/marian-dev`.

Junczys-Dowmunt, Marcin, Kenneth Heafield, Hieu Hoang, Roman Grundkiewicz, and Anthony Aue. 2018b. Marian: Cost-effective high-quality neural machine translation in C++. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 129–135, Melbourne, Australia. Association for Computational Linguistics.

Kim, Yoon and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327, Austin, Texas. Association for Computational Linguistics.

Kim, Young Jin, Marcin Junczys-Dowmunt, Hany Hassan, Alham Fikri Aji, Kenneth Heafield, Roman Grundkiewicz, and Nikolay Bogoychev. 2019. From research to production and back: Ludicrously fast neural machine translation. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 280–288, Hong Kong. Association for Computational Linguistics.

Kocmi, Tom, Martin Popel, and Ondrej Bojar. 2020. Announcing CzEng 2.0 parallel corpus with over 2 gigawords. *arXiv preprint arXiv:2007.03006*.

Koehn, Philipp, Franz J. Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 127–133.

Kudo, Taku and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics. Code at `https://github.com/google/sentencepiece`.

Liu, Lemao, Masao Utiyama, Andrew Finch, and Eiichiro Sumita. 2016. Neural machine translation with supervised attention. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3093–3102, Osaka, Japan. The COLING 2016 Organizing Committee.

Popel, Martin. 2018. CUNI transformer neural MT system for WMT18. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 482–487, Belgium, Brussels. Association for Computational Linguistics.

Popel, Martin. 2020. CUNI English-Czech and English-Polish Systems in WMT20: Robust document-level training. In *Proceedings of the Fifth Conference on Machine Translation: Shared Task Papers*, Online. Association for Computational Linguistics.

Popel, Martin, Marketa Tomkova, Jakub Tomek, Łukasz Kaiser, Jakob Uszkoreit, Ondřej Bojar, and Zdeněk Žabokrtskỳ. 2020. Transforming machine translation: a deep learning system reaches news translation quality comparable to human professionals. *Nature Communications*, 11(1):1–15.

Post, Matt. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

Sennrich, Rico, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Vaswani, Ashish, Samy Bengio, Eugene Brevdo, Francois Chollet, Aidan N. Gomez, Stephan Gouws, Llion Jones, Łukasz Kaiser, Nal Kalchbrenner, Niki Parmar, Ryan Sepassi, Noam Shazeer, and Jakob Uszkoreit. 2018. Tensor2tensor for neural machine translation. *CoRR*, abs/1803.07416.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

The University of Edinburgh's submission to the German-to-English and English-to-German Tracks in the WMT 2020 News Translation and Zero-shot Translation Robustness Tasks

Ulrich Germann
University of Edinburgh, UK
ugermann@inf.ed.ac.uk

Abstract

This paper describes the University of Edinburgh's Submission of German ↔ English systems to the WMT2020 Shared Tasks on News Translation and Zero-shot Robustness.

1 Introduction

This paper describes the University of Edinburgh's submission to the German-to-English and English-to-German tracks in the WMT 2020 News Translation and Zero-shot Robust translation tasks. We built our systems in three stages, loosely following the procedure by Junczys-Dowmunt (2018b). All translation models mentioned in this paper were trained with the Marian toolkit (Junczys-Dowmunt et al., 2018).

2 3-stage Build Process

We distinguish three types of training data provided for the tasks, shown in Table 1.

Table 1: Data available for training systems

corpus	sentence(pair)s
High-quality parallel data	
Europarl	ca. 1.79 M
Rapid	ca. 1.45 M
News Commentary	ca. 0.35 M
Crawled parallel data	
ParaCrawl 5.1	ca. 34.37 M
CommonCrawl	ca. 2.40 M
WikiMatrix	ca. 6.22 M
WikiTitles	ca. 1.38 M
Monolingual crawled news data	
German	ca. 327.69 M
English	ca. 233.50 M

2.1 Round 1: Training models for scoring crawled parallel data

In the first round, we trained base transformer models (Vaswani et al., 2017) with Sentence-Piece subword segmentation (Kudo and Richardson, 2018) with a joint vocabulary of 32K tokens on the high-quality parallel data. The joint vocabulary remains the same for all models. We applied a binomial sentence length model to remove from the parallel data sentence pairs with an unreasonable sentence length ratio. The model assumes that a pair of sentences of lengths K and L is produced by a series of K+L flips of a biased coin. The bias is based on the corpus-level ratio of tokens; for German and English, we determined that there are on average 1.0723 English tokens per German token, so the Null Hypothesis assumes that an English word is generated with a probability of 51.75%, and a German word with a probability of 48.25%. For each sentence pair, we determine the p-value of the Null Hypothesis; if it is less than 0.5%, the sentence pair is discarded. This sentence filtering filtered out less than 2% of the EuroParl data, ca 3.4% of the NewsCommentary data, and 11% of the Rapid corpus. The numbers given in Table 1 are after filtering.

2.2 Selecting crawled parallel data

We used these models to compute the length-normalized cross-entropy for each sentence pair in the available crawled parallel data in both translation directions, and combined these two entropies into the dual cross-entropy score (Junczys-Dowmunt, 2018a). To bias data selection towards the news domain, we also computed length-normalized cross-entropy for each sentence with a 5-gram language model[1] trained on the respective monolingual news data for the target side and added the two scores to obtain a single score for ranking candidates. The top n candidates from the crawled parallel data were pooled with the high-quality parallel data for the second round of training. We experimented with the top 15 M, top 20 M, and top 25 M candidates from the pool of crawled parallel data. We did not put effort into cleaning or filtering the data prior to scoring, as we assumed that poor candidates would be detected by the dual cross-entropy score.

[1] https://github.com/kpu/kenlm

Proceedings of the 5th Conference on Machine Translation (WMT), pages 197–201
Online, November 19–20, 2020.

For the German-to-English system, we unfortunately committed a serious blunder that we did not notice until shortly before submitting this system description: instead of sorting in descending order of ranking score, we accidentally sorted in descending order of provenance label first and (also lexicographically) ranking score second, so that for the translation direction German→English, the crawled data selection contained all of WikiMatrix data, none of the CommonCrawl data, and a selection of ParaCrawl data. The selection error rate in the Top-25M confiuration is ca 66% (i.e., 66% of that data should not have been selected, and we missed 66% of the data that we wanted to select). As we used Round2-models for back-translation of monolingual data, this error may have also tainted the training of the English-to-German system.

2.3 Round 2: Big transformers for back-translation

We then trained big transformer models for back-translation of monolingual news data, using the "top" (see above for our blunder on German-to-English data selection) 25M candidates.

2.4 Back-translation of news data

We used single models to translate all of the news data for German and English, adding a bit of noise in the translation by adding Gumbel noise to the output layer, thus adding some randomness to the translation process.

2.5 Round 3: Training final models with back-translation.

In the final round of training, we trained big transformer models on a blend of back-translated data (75%), crawled parallel data (15%) and high-quality parallel data (10%). Due to the volume of training data available for this round, we replaced full shuffling of the data for each epoch by random selection: we loop over each data source, fully shuffling the latter two data sources (crawled and high-quality parallel) in each iteration, but shuffling the backtranslated news data only once and then randomly selecting only 10% of the data in each iteration. Reading separately from the three data sources, the data feeder randomly selects one data source at a time according to the aforementioned distribution of 75,15, and 10% and outputs the next sentence pair in the queue.

3 Training details

For training, we experimented with variations on learning rate, batch size, warmup, and norm clipping. Due to an apparent bug in the implementation of norm clipping in Marian,[2] the Marian authors do currently not recommend to use norm clipping with Marian.[3] However, we found that without it, training would occasionally fail due to exploding gradients. Norm clipping also allowed us to be more agressive with the learning rate,

Settings and BLEU scores[4] on the validation set (WMT19) are shown in Table 2. Effective batch size was influenced by the GPU memory allocated and the number of gradients accumulated before a parameter update ("optimizer delay"). In principle, doubling the optimizer delay should double the batch size, but we found this not to be the case in practice. An analysis after the fact revealed that this was due to interactions between automatically fitting batch sizes to available memory (`--maxi-batch-fit`), setting maxibatch size, and the optimizer delay parameter that are currently not documented well for the Marian toolkit and that we misunderstood.

For German-to-English, training on back-translated data did not lead to improvements in terms of BLEU on the validation set, so we ensembled the 4 listed round-2 models for our primary submission. We were able to boost the BLEU score of the raw translation output by 1.3 BLEU points with a simple post-processing step that simply adjusts quotations marks to German spelling conventions.

For English-to-German, we submitted an ensemble of the 8 round-2 models with the highest BLEU score with respect to the validation set (WMT19).

Here, too, training on back-translated news data did not lead to an improvement over the best Round-2 model, so that we did not use any round-3 model for the final submission. We were able to boost performance by increasing the batch size during training, which is in line with our findings from last year (Bawden et al., 2019), but the effect was much smaller this year. This may be due to the fact that the initial model (#10, English→German) was already trained with a fairly large batch size.

4 Results

Table 3 shows the overlap (as measured in BLEU) for all primary systems submissions to the News Translation Task as released by the workshop organizers. We notice a few things. First, our data selection blunder for the German-to-English system has not catastrophically harmed final performance. In fact, in terms of ranking with respect to BLEU, our German-to-English system does better than our English-to-German system.

[2] Gradients aren't normalized but norm clipping isn't adjusted for batch size.

[3] Personal communication with R. Grundkiewicz.

[4] All BLEU scores reported in this paper were computed with SacreBLEU (Post, 2018); BLEU+case.mixed+numrefs.1+smooth.exp+tok.13a +version.1.4.14.

Table 2: Training details. See Section 3 for details.

run	cont. from	WMT19 BLEU	transformer type	batch tokens	learning rate	clip norm	warmup update	crawled data
Round 1 German→ English								
1		29.6	basic	ca. 119K	0.0009	5	16K	—
Round 2 German→ English								
2		41.02	big	ca. 84K	0.0009	5	16K	20M
3		41.21	big	ca. 31K	0.0002	0	8K	20M
4		41.22	big	ca. 81K	0.0003	0	16K	20M
5		41.47	big		0.0003	0	16K	25M
Round 3 German→ English								
6		39.51	big	ca. 124K	0.0002	0	8K	25M
7	5	41.04	big	ca. 246K	0.0003	5	1K	25M
Round 1 English→ German								
1		TBD	basic	ca. 119K	0.0009	5	16K	—
Round 2 English→ German								
2		41.63	big	ca. 20K	0.0002	0	8K	20M
3		41.73	big	ca. 83K	0.0003	0	16K	25M
4		41.85	big	ca. 103K	0.0002	0	8K	20M
5		41.89	big	ca. 185K	0.0009	0	26K	20M
6		42.02	big	ca. 34K	0.0002	0	8K	20M
7		42.13	big	ca. 144K	0.0003	0	16K	25M
8		42.49	big	ca. 26K	0.0009	5	16K	15M
9		42.62	big	ca. 56K	0.0002	0	8K	20M
Round 3 English→ German								
10		31.23	big	ca. 120K	0.0003	0	8K	25M
11	10	41.46	big	ca. 120K	0.0002	5	–	25M
12	11	42.01	big	ca. 205K	0.0002	5	–	25M
13	12	42.61	big	ca. 334K	0.0002	0	–	25M
13	12	41.94	big	ca. 339K	0.0002	0	–	25M

Table 3: Overlap between references and submissions for German-to-English (top) and English-to-German (bottom), measured in BLEU, with "references" in the columns and "candidates" in the rows.

	REF1	Tohoku...	Huoshan...	OPPO	UEDIN	Online-B	Online-G	Online-A	PROMT...	Online-Z	REF2	Biomed...	zlabs-nlp	yolo
REF1	100.00	43.80	43.50	43.20	42.30	41.90	41.40	40.40	39.60	35.40	34.00	32.10	31.50	0.20
Tohoku...	43.80	100.00	77.00	71.80	70.90	63.90	69.30	66.60	64.10	54.20	42.20	47.40	47.10	0.20
Huoshan...	43.50	77.10	100.00	74.60	73.60	65.50	70.00	70.40	68.50	57.40	42.20	48.80	47.80	0.20
OPPO	43.20	71.90	74.60	100.00	71.10	64.50	66.80	63.90	63.50	55.10	41.90	48.10	47.00	0.30
UEDIN	42.30	70.90	73.60	71.10	100.00	64.90	69.60	67.40	67.60	57.00	40.80	50.10	50.20	0.20
Online-B	41.90	64.00	65.50	64.50	64.90	100.00	61.90	61.70	61.20	54.10	40.90	45.70	44.80	0.20
Online-G	41.40	69.30	70.00	66.80	69.70	61.90	100.00	64.70	63.90	53.80	39.80	46.20	45.50	0.20
Online-A	40.40	66.60	70.40	63.90	67.40	61.70	64.70	100.00	64.30	53.70	39.60	46.10	44.50	0.20
PROMT...	39.60	64.20	68.50	63.50	67.60	61.20	63.90	64.30	100.00	56.80	38.60	49.20	47.30	0.20
Online-Z	35.40	54.20	57.40	55.10	57.00	54.10	53.80	53.70	56.80	100.00	35.00	43.00	43.10	0.20
REF2	34.00	42.20	42.20	41.90	40.90	40.90	39.80	39.60	38.60	35.00	100.00	31.00	30.50	0.20
Biomed...	32.10	47.40	48.80	48.10	50.10	45.70	46.30	46.10	49.20	42.90	31.00	100.00	46.80	0.20
zlabs-nlp	31.50	47.10	47.80	47.00	50.20	44.70	45.50	44.40	47.30	43.10	30.50	46.80	100.00	0.30
yolo	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.30	100.00

	REF1	Tohoku...	Tencent...	OPPO	Huoshan...	eTranslation	Online-B	UEDIN	Online-A	AFRL	REF2	PROMT...	Online-Z	zlabs-nlp	Online-G	Biomed...
REF1	100.00	38.70	38.60	38.50	38.10	37.90	36.40	36.30	36.10	34.10	32.50	31.80	29.60	28.10	27.20	25.20
Tohoku...	38.80	100.00	70.40	69.80	71.50	70.30	63.20	65.30	67.50	59.90	38.80	60.10	50.90	48.20	50.60	41.10
Tencent...	38.60	70.40	100.00	74.30	74.20	70.40	59.60	67.80	63.80	62.10	38.00	59.00	51.20	45.80	49.00	41.30
OPPO	38.60	69.80	74.30	100.00	75.20	72.90	60.60	68.00	63.60	63.70	38.00	58.00	50.10	45.00	48.40	40.10
Huoshan...	38.20	71.60	74.20	75.10	100.00	72.80	60.40	69.50	64.30	62.50	37.90	60.50	51.60	46.40	49.10	41.30
eTranslation	37.90	70.30	70.40	72.90	72.80	100.00	58.20	68.80	63.90	64.30	37.60	57.60	49.10	45.10	48.70	39.80
Online-B	36.50	63.20	59.50	60.50	60.40	58.10	100.00	57.00	62.40	53.60	36.60	56.10	48.50	46.00	47.80	41.80
UEDIN	36.30	65.30	67.80	68.00	69.40	68.80	57.10	100.00	62.50	61.70	35.50	60.90	51.50	48.70	50.10	42.00
Online-A	36.20	67.50	63.80	63.50	64.20	63.90	62.40	62.40	100.00	55.70	36.10	60.50	49.80	48.40	50.80	42.10
AFRL	34.10	59.90	62.20	63.80	62.60	64.40	53.70	61.70	55.80	100.00	34.20	52.60	46.60	43.60	45.50	39.70
REF2	32.50	38.80	37.90	38.00	37.90	37.60	36.50	35.40	36.10	34.20	100.00	31.60	28.70	27.40	27.00	24.50
PROMT...	31.80	60.20	59.00	58.00	60.50	57.50	56.20	60.90	60.50	52.50	31.70	100.00	50.60	47.70	51.60	43.20
Online-Z	29.60	51.00	51.10	50.10	51.60	49.10	48.50	51.50	49.80	46.60	28.70	50.60	100.00	42.00	43.20	38.00
zlabs-nlp	28.20	48.20	45.80	44.90	46.40	45.10	46.00	48.70	48.40	43.50	27.40	47.70	42.00	100.00	40.10	40.00
Online-G	27.20	50.70	49.00	48.50	49.10	48.70	47.90	50.20	50.80	45.40	27.00	51.60	43.20	40.10	100.00	37.20
WMTBiomedBaseline	25.20	41.10	41.30	40.20	41.30	39.80	41.80	42.00	42.10	39.70	24.50	43.20	38.00	40.00	37.30	100.00

Table 4: BLEU scores for the Zero-shot Robustness Task

test set	BLEU
en→de set 1	35.1
de→en set 1	38.8
de→en set 3	43.8

Second, the difference between the independently created human reference translations (REF1 and REF2) is larger than the difference between the top-performing automatic translations and either of the systems. We conjecture that is is due to the fact that individual translators will have individual translation styles, whereas automatically trained systems learn to emulate the "average" translator. However, this once again raises the question about the validity of BLEU as a measure of translation quality.

Third, we find the high overlap between the top-scoring automatic systems remarkable. This suggests that under the constrained conditions, independently systems do learn a very similar style of translation.

For the Zero-shot Robustness Task, we used the same systems as for for the News Translation Task. We report BLEU scores for the Zero-Shot Robustness Task in Table 4.

Acknowledgements

This work was supported by funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825303 (Bergamot) and 825627 (European Language Grid).

This work was performed using resources provided by the Cambridge Service for Data Driven Discovery (CSD3) operated by the University of Cambridge Research Computing Service (http://www.csd3.cam.ac.uk/), provided by Dell EMC and Intel using Tier-2 funding from the Engineering and Physical Sciences Research Council (capital grant EP/P020259/1), and DiRAC funding from the Science and Technology Facilities Council (www.dirac.ac.uk).

References

Bawden, Rachel, Nikolay Bogoychev, Ulrich Germann, Roman Grundkiewicz, Faheem Kirefu, Antonio Valerio Miceli Barone, and Alexandra Birch. 2019. The university of Edinburgh's submissions to the WMT19 news translation task. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 103–115, Florence, Italy. Association for Computational Linguistics.

Junczys-Dowmunt, Marcin. 2018a. Dual conditional cross-entropy filtering of noisy parallel corpora. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 888–895, Belgium, Brussels. Association for Computational Linguistics.

Junczys-Dowmunt, Marcin. 2018b. Microsoft's submission to the WMT2018 news translation task: How I learned to stop worrying and love the data. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 425–430, Belgium, Brussels. Association for Computational Linguistics.

Junczys-Dowmunt, Marcin, Kenneth Heafield, Hieu Hoang, Roman Grundkiewicz, and Anthony Aue. 2018. Marian: Cost-effective high-quality neural machine translation in C++. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 129–135, Melbourne, Australia. Association for Computational Linguistics.

Kudo, Taku and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics. Code at `https://github.com/google/sentencepiece`.

Post, Matt. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Contact Relatedness can help improve multilingual NMT: Microsoft STCI-MT @ WMT20

Vikrant Goyal, Anoop Kunchukuttan, Rahul Kejriwal, Siddharth Jain, Amit Bhagwat
STC India, Microsoft, Hyderabad
{vikgoyal, ankunchu, rakejriw, sija, amitb}@microsoft.com

Abstract

We describe our submission for the English→Tamil and Tamil→English news translation shared task. In this submission, we focus on exploring if a low-resource language (Tamil) can benefit from a high-resource language (Hindi) with which it shares contact relatedness. We show utilizing contact relatedness via multilingual NMT can significantly improve translation quality for English-Tamil translation.

1 Introduction

In recent years, Neural Machine Translation (Luong et al., 2015; Bahdanau et al., 2015; Johnson et al., 2017; Wu et al., 2018; Vaswani et al., 2017) (NMT) has become the most prominent approach to Machine Translation (MT) due to its simplicity, generality and effectiveness. In NMT, a single neural network often consisting of an encoder and a decoder is used to directly maximize the conditional probabilities of target sentences given the source sentences in an end-to-end paradigm. NMT models have been shown to surpass the performance of previously dominant statistical machine translation (SMT) (Koehn, 2009) on many well-established translation tasks. However, in order to obtain good translation quality, NMT systems tend to require very large parallel training corpora (Koehn and Knowles, 2017). Such corpora are not yet available for many language pairs.

The Indian subcontinent forms a *linguistic area* where languages from the Dravidian and Indo-Aryan families have been in **contact** for a long time leading to significant sharing of vocabulary and a convergence of linguistic features (Emeneau, 1956). Tamil is a major language from the Dravidian language family spoken in Southern India while Hindi is a widely spoken Indo-Aryan language. Kunchukuttan and Bhattacharyya (2020) estimate that lexical similarity between Hindi and Tamil to be around 27% in terms of character LCSR (Melamed, 1995), while multiple works have shown that language representations of Tamil and Hindi cluster in the same neighbourhood in a multilingual vector space (Kudugunta et al., 2019; Oncevay et al., 2020).

While English-Tamil parallel corpora is limited, more parallel corpora is available for English-Hindi. In this paper, we explore if English-Hindi can improve English-Tamil machine translation due to the similarities between Hindi and Tamil on account of contact relatedness. To this end, we train multilingual NMT models for English-Hindi and English Tamil (and vice-versa). Previous work has explored whether high-resource languages can transfer knowledge to genetically-related low-resource languages (Nguyen and Chiang, 2017; Dabre et al., 2017). In contrast, we explore if **contact relatedness** can benefit low resource languages. We further explore if reducing the divergence between Tamil and Hindi data by representing them in the same script is beneficial. In addition, we explored target agreement models, tagged and noisy back-translation in our submission.

2 Neural Machine Translation

Given a bilingual sentence pair (x, y), an NMT model learns its parameters θ by maximizing the log-likelihood $P(y|x;\theta)$, which is usually decomposed into the product of the conditional probability of each target word: $P(y|x;\theta) = \prod_{t=1}^{m} P_\theta(y_t|y_1, y_2, .., y_{t-1}, x;\theta)$, where m is the length of sentence y.

An encoder-decoder framework (Bahdanau et al., 2015; Luong et al., 2015; Gehring et al., 2017; Vaswani et al., 2017) is usually adopted to model the conditional probability $P(y|x;\theta)$. The encoder maps the input sentence x into a set of hidden rep-

Proceedings of the 5th Conference on Machine Translation (WMT), pages 202–206
Online, November 19–20, 2020.

resentations h, and the decoder generates the target token y_t at position t using the previously generated target tokens $y_{<t}$ and the source representations h. Both the encoder and decoder can be implemented by different structure of neural models, such as RNN (LSTM/GRU) (Bahdanau et al., 2015; Luong et al., 2015), CNN (Gehring et al., 2017) and self-attention (Vaswani et al., 2017). Besides the basic component of the encoder and decoder, a source-target attention mechanism (Bahdanau et al., 2015) is usually adopted to selectively focus on the source representations when generating a target token.

The Transformer (Vaswani et al., 2017) model is the state-of-the-art NMT model relying completely on self-attention mechanism to compute representations of its input and output without using recurrent neural networks (RNN) or convolutional neural networks (CNN). In this work, we use the Transformer architecture in all of our NMT models. We use smaller capacity networks compared to *Transformer-base*, given the smaller size of the parallel data.

3 Multilingual Learning for NMT

The objective of multilingual learning for NMT is to construct a single model for translating to and from multiple languages. Multilingual models can help improve performance of low-resource languages by transferring from high-resource related languages they are trained jointly with. Firat et al. (2017) introduced a many-to-many system, which still relied upon separate encoders and decoders for each language along with a shared attention mechanism. In contrast, Johnson et al. (2017) introduced a "language flag"-based approach that shares the attention mechanism and a single encoder-decoder network to enable multilingual models. A language flag or token is part to the input sequence to indicate which direction to translate to. The decoder learns to generate the target given this input. This approach has been shown to be simple and effective and we use this in our multilingual models. As mentioned earlier, we train a joint Hindi,Tamil to English model as well as a joint English to Hindi,Tamil model. Our hypothesis is that the contact relatedness between Hindi and Tamil will help transfer knowledge from Hindi to Tamil effectively.

Tamil and Hindi use different scripts. However, it is possible to map almost all Devanagari (the script used for Hindi) characters to Tamil characters. This mapping is deterministic but lossy since the Tamil character set is smaller than the Devanagari character set. Such mapping will help to utilize the lexical similarity between the two languages directly. Hence, we convert all Hindi data to Tamil script during a pre-processing step. We also report results of our MultiNMT models without script conversion of Hindi to Tamil.

4 Backtranslation

Backtranslation (BT) (Sennrich et al., 2016a) is a widely used data augmentation method where the reverse direction is used to translate sentences from target-side monolingual data into the source language. This synthetic parallel data is combined with the actual parallel data to re-train the model leading to better language modelling on the target-side, regularization and target domain adaptation. Backtranslation is particularly useful for low-resource languages. We use backtranslation to augment our multilingual models. The backtranslation data is generated by multilingual models in the reverse direction, hence some implicit multilingual transfer is incorporated in the backtranslated data also.

4.1 Noisy and Tagged Backtranslation

Backtranslation typically uses beam search (Sennrich et al., 2016a) or just greedy search (Lample et al., 2018a,b) to generate synthetic source sentences. Both are approximate algorithms to identify the maximum a-posteriori (MAP) output, i.e. the sentence with the largest estimated probability given an input. Beam is generally successful in finding high probability outputs (Ott et al., 2018). However, MAP prediction can lead to less rich translations since it always favors the most likely alternative in case of ambiguity. Edunov et al. (2018) argue that this is also problematic for a data augmentation scheme such as backtranslation. Beam and greedy search focus on the head of the model distribution which results in very regular synthetic source sentences that do not properly cover the true data distribution. Following the approach proposed by Edunov et al. (2018), we apply noising to the beam search outputs. In particular, we transform source sentences with three types of noise: deleting words with probability 0.1, replacing words by a filler token with probability 0.1, and swapping words which is implemented as a random permutation over the tokens, drawn from the uniform distribution but restricted to swapping words no

further than three positions apart.

Caswell et al. (2019) showed that main purpose of the synthetic noise is not to diversify the source but simply to indicate that the given source is synthetic. They proposed to prepend the input sequences of the synthetic data with a reserved token like <BT> to indicate that the given source is synthetic. In this paper, we experiment with both Noisy BT and Tagged BT.

5 Target Agreement

Due to the autoregressive structure, current NMT systems usually suffer from the so-called exposure bias problem (Bengio et al., 2015): during inference, true previous target tokens are unavailable and replaced by tokens generated by the model itself, thus mistakes made early can mislead subsequent translation, yielding unsatisfactory translations with good prefixes but bad suffixes. Such an issue can become severe as sequence length increases. Zhang et al. (2019) showed that the impact of this can be reduced by augmenting the training data with synthetic targets generated by a left-to-right (L2R) and a right-to-left (R2L) translation model. The directionality of the synthetic targets ensures that decoder input distribution becomes noisier (as happens at runtime) along one side of the target. The augmented data thus serves to reduce the divergence in the decoder input distribution. This is especially relevant to low-resource language scenarios where the model is not as robust to the decoder input distribution.

6 Experimental Settings

6.1 Dataset

We train our models only on the parallel data provided for the task (see Table 1 for dataset details). For backtraslation, we randomly selected 10M sentences from the newscrawl 2019 English monolingual corpora. For Tamil monolingual corpora, used the entire newscrawl corpus (0.7M), Wikipedia corpus and part of the CommonCrawl data made available for a consolidated corpus of 10M sentences. We use IIT-Bombay Hindi-English parallel corpora v2.0 (Kunchukuttan et al., 2018) containing 1.5M parallel sentences to build our multilingual models. We used UFAL's Tamil-English dev set containing 1,000 parallel sentences for tuning our models.

Dataset	# of Sentences
Wikititles	102,146
Wikimatrix	52,669
PMIndia	39,526
Tanzil (Koran)	93,540
NLPC_UOM	8,945
PIB (CVIT@IIITH)	60,836
MKB (CVIT@IIITH)	5,744
UFAL	166,871
Total	530,277

Table 1: Tamil-English parallel corpus statistics.

6.2 Data Processing

We use the Moses (Koehn et al., 2007) toolkit[1] for lowercasing, tokenization and cleaning the English side of the data. Both Tamil and Hindi data are first normalized and then tokenized. The Hindi data is mapped to Tamil script. We use the Indic NLP library[2] (Kunchukuttan, 2020) for text processing of the Indic languages. We remove all sentences of length greater than 80 words from our training corpus. In all cases, we use BPE subword segmentation (Sennrich et al., 2016b) with 32k merge operations. In case of mulitlingual models, we learn the BPE vocabulary jointly on the Hindi and Tamil data.

6.3 Training Details

For all of our experiments, we use the fairseq (Ott et al., 2019) toolkit[3]. We use the Transformer model with 4 layers in both the encoder and decoder, each with 512 hidden units. The word embedding size is set to 512 and 8 attention heads are used. The training is done in batches of maximum 2048 tokens at a time with dropout set to 0.2. We use the Adam (Kingma and Ba, 2015) optimizer to optimize model parameters with $\beta_1 = 0.9$, $\beta_2 = 0.98$ and $\epsilon = 1e - 9$ and we use the same learning rate schedule as Vaswani et al. (2017). We validate the model after each epoch via label smoothed cross entropy loss and perplexity on the development set. We train all our NMT models till convergence where convergence is determined by label smoothed cross entropy loss on the development set. After translation at the test time, we rejoin the translated BPE segments. Finally,

[1] https://github.com/moses-smt/mosesdecoder
[2] https://anoopkunchukuttan.github.io/indic_nlp_library/
[3] https://github.com/pytorch/fairseq

we evaluate the accuracy of our translation models using SacreBLEU (Post, 2018).

7 Results and Discussion

We report the SacreBLEU scores on the dev sets and test sets provided in the WMT20 News translation task for both the language directions: Tamil→English (Table 2) and English→Tamil (Table 3). We experimented with Multilingual (MultiNMT), Backtranslation (BT), Noisy Backtranslation (noisyBT) and Tagged Backtranslation (taggedBT) and Target Agreement (TA) models.

We observe that our multilingual models outperform the baseline bilingual models by significant margins. On the newstest2020 set, we see an improve of 2.9 and 1.5 BLEU points respectively in the Tamil→English and English→Tamil directions respectively. Note that the gains translating into multiple targets is lower. However, when backtranslated data is added we observe an improvement of 2.3 BLEU points in the en→ta quality. Note that the backtranslation data was generated via the multilingual ta→en model, hence there is an implicit benefit from multilinguality when using backtranslation. Backtranslation also provides a good improvement in the en→ta. We see that representing Indian language data in the same script was very beneficial for ta→en translation, while it did not help in the other direction. Having disjoint vocabularies during generation possibly helps the model learn distinct language models for Hindi and Tamil.

Our results indicate that Target Agreement and Noisy and Tagged Backtranslation schemes are not helpful in increasing the translation performance of the NMT models for the language pairs of our interest and requires more investigation on low resource language translation tasks. Further analysis is needed to understand why backtranslation variants and target agreement did not show improvements in our setting.

System	newsdev2020	newstest2020
Transformer baseline	10.4	10.0
MultiNMT (no script conversion)	12.5	12.2
MultiNMT	13.0	12.9
MultiNMT+BT	**19.1**	**14.2**
MultiNMT+noisyBT	18.3	14.2
MultiNMT+taggedBT	17.6	13.5
MultiNMT+BT+TA	19.1	14.2

Table 2: Tamil→English (Ta-En) experiment results.

System	newsdev2020	newstest2020
Transformer baseline	6.1	3.5
MultiNMT (no script conversion)	7.5	4.9
MultiNMT	7.4	5.0
MultiNMT+BT	**11.5**	**7.3**
MultiNMT+BT+TA	11.3	7.3

Table 3: English→Tamil (En-Ta) experiment results.

8 Conclusion

We believe contact relatedness can be utilized in the multilingual NMT framework for improving low-resource language translation. Our initial results confirm this for English-Tamil translation aided by English-Hindi data. In addition, we show, that the popular data augmentation methods like backtranslation further helps in increasing the translation performance of Multilingual NMT models.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *ICML*.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems*, pages 1171–1179.

Isaac Caswell, Ciprian Chelba, and David Grangier. 2019. Tagged back-translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 53–63.

Raj Dabre, Tetsuji Nakagawa, and Hideto Kazawa. 2017. An Empirical Study of Language Relatedness for Transfer Learning in Neural Machine Translation. In *The 31st Pacific Asia Conference on Language, Information and Computation*.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500.

Murray B Emeneau. 1956. India as a lingustic area. *Language*.

Orhan Firat, Kyunghyun Cho, Baskaran Sankaran, Fatos T Yarman Vural, and Yoshua Bengio. 2017. Multi-way, multilingual neural machine translation. *Computer Speech and Language*, 45(C):236–252.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1243–1252. JMLR. org.

Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICML*.

Philipp Koehn. 2009. *Statistical machine translation*. Cambridge University Press.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*, pages 177–180. Association for Computational Linguistics.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39.

Sneha Kudugunta, Ankur Bapna, Isaac Caswell, and Orhan Firat. 2019. Investigating multilingual NMT representations at scale. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1565–1575, Hong Kong, China. Association for Computational Linguistics.

Anoop Kunchukuttan. 2020. The IndicNLP Library. https://github.com/anoopkunchukuttan/indic_nlp_library/blob/master/docs/indicnlp.pdf.

Anoop Kunchukuttan and Pushpak Bhattacharyya. 2020. Utilizing Language Relatedness to improve Machine Translation: A Case Study on Languages of the Indian Subcontinent. *arxiv pre-print 2003.08925*.

Anoop Kunchukuttan, Pratik Mehta, and Pushpak Bhattacharyya. 2018. The IIT Bombay English-Hindi Parallel Corpus. In *LREC*.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018a. Unsupervised machine translation using monolingual corpora only. In *International Conference on Learning Representations*.

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018b. Phrase-based & neural unsupervised machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5039–5049.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421.

I Dan Melamed. 1995. Automatic evaluation and uniform filter cascades for inducing n-best translation lexicons. In *Third Workshop on Very Large Corpora*.

Toan Q Nguyen and David Chiang. 2017. Transfer Learning across Low-Resource, Related Languages for Neural Machine Translation. In *International Joint Conference on Natural Language Processing*.

Arturo Oncevay, Barry Haddow, and Alexandra Birch. 2020. Bridging linguistic typology and multilingual machine translation with multi-view language representations. *arxiv pre-print 2004.14923*.

Myle Ott, Michael Auli, David Grangier, and Marc'Aurelio Ranzato. 2018. Analyzing uncertainty in neural machine translation. In *ICML*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Matt Post. 2018. A call for clarity in reporting bleu scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Lijun Wu, Yingce Xia, Fei Tian, Li Zhao, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. 2018. Adversarial neural machine translation. In *Asian Conference on Machine Learning*, pages 534–549.

Zhirui Zhang, Shuangzhi Wu, Shujie Liu, Mu Li, Ming Zhou, and Tong Xu. 2019. Regularizing neural machine translation by target-bidirectional agreement. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 443–450.

The AFRL WMT20 News-Translation Systems

Jeremy Gwinnup and **Timothy Anderson**
Air Force Research Laboratory
{jeremy.gwinnup.1, timothy.anderson.20}@us.af.mil

Abstract

This report summarizes the Air Force Research Laboratory (AFRL) machine translation (MT) systems submitted to the news-translation task as part of the 2020 Conference on Machine Translation (WMT20) evaluation campaign. This year we largely repurpose strategies from previous years' efforts with larger datasets and also train models with precomputed word alignments under various settings in an effort to improve translation quality.

1 Introduction

As part of the 2020 Conference on Machine Translation (wmt, 2020) news-translation shared task, the AFRL human language technology team participated in the Russian–English portion of the competition. We largely employed our strategies from last year including language-based filtering of training corpora with fastText (Joulin et al., 2016b,a), employing transformer-based (Vaswani et al., 2017) translation models and once again utlitizing system combination to fuse outputs from OpenNMT (Klein et al., 2018), Marian (Junczys-Dowmunt et al., 2018) and Moses (Koehn et al., 2007) systems. We also examine the effects of training Marian models with externally generated word alignments as described in (Alkhouli et al., 2018).

2 Data processing

For purposes of training our systems, we use the following parallel corpora: Commoncrawl (Smith et al., 2013), Yandex[1], UN v1.0 (Ziemski et al., 2016), Paracrawl[2](Esplà et al., 2019), Wikimatrix (Schwenk et al., 2019), and backtranslated data from our WMT17 system (Gwinnup et al., 2017) as well as Edinburgh's WMT17 system (Sennrich et al., 2017) yielding a raw corpus of over 76.3 million lines.

We prepare training corpora in a similar manner described in (Gwinnup et al., 2018), however this year, we utilize SentencePiece (Kudo and Richardson, 2018) with a 46k-entry vocabulary[3] for processing subword units instead of byte-pair encoding (BPE) (Sennrich et al., 2016).

2.1 Language-ID based data filtering

As with last year's efforts, we again employ fastText (Joulin et al., 2016b,a) to filter the various parallel corpora with a utility examining the source and target sentence pairs, discarding pairs where either (or both) sentence in the pair falls below a threshold score of 0.8. We wished to explore different threshold values, but our team did not have access to the majority of our computational assets due to the COVID-19 pandemic, limiting the bandwidth available for experiments.

We show the results of language-ID based filtering in Table 1. On average, 76.79% of the original training data is retained, with our WMT17 backtranslated data retaining the largest percentage of lines at 93.22% - this is interesting since that data originated as English and was translated to Russian with a very shallow Amun (Hoang et al., 2018) model. Again, Paracrawl yielded the least percentage of retained lines at 42.90%, but is understandable due to the "raw" nature of this particular release.

2.2 Guided Alignment

Inspired by the results in (Alkhouli et al., 2018), we've examined effects of using precomputed word alignments as a guide during training; Marian has a facility to train in this manner. Alignments were generated using Fastalign (Dyer et al., 2013)

[1] https://translate.yandex.ru/corpus?lang=en
[2] Version 1 Russian–English parallel data
[3] This vocabulary size performed best in empirical testing in our WMT19 submission.

Proceedings of the 5th Conference on Machine Translation (WMT), pages 207–212

corpus	unfiltered lines	filtered lines	percent remain
commoncrawl	723,256	655,069	90.57%
news-commentary-v15	319,242	286,947	89.88%
yandex	1,000,000	901,318	90.13%
un-2016	11,365,709	9,871,406	86.85%
paracrawl	12,061,155	5,173,675	42.90%
wikimatrix	5,203,872	4,287,881	82.40%
wmt17-afrl-bt	8,921,942	8,317,107	93.22%
wmt17-uedin-bt	36,772,770	29,074,022	79.06%
Total	76,367,946	58,567,425	76.69%

Table 1: Results of language-id based Russian–English corpus filtering with threshold of 0.8

on both "plain" and SentencePiece-processed data; MGIZA (Gao and Vogel, 2008) alignments were only generated for the word-based data. In order to generate these alignments, the language-id filtered corpus described in the previous section was further processed using Moses's clean-corpus-n-ratio script as well as escaping various characters and entities (such as ' replaced with &) yielding a final corpus of 49,866,140 lines. Additionally, a 46k entry SentencePiece model is built on this corpus with user-defined vocabularies for the tokens escaped during processing.

We use a Procrustes alignment projection script[4] to effectively map alignments generated on whole word tokens to the equivalent series of subword tokens in the SentencePiece processed data. Comparisons are drawn between Marian models trained on these various conditions in Section 3.2.

3 Machine Translation Systems

This year, we focused system-building efforts on the OpenNMT, Marian, and Moses toolkits. While most of our experimentation builds off of previous years' efforts, this year we examine the effects of "guided-alignment" training with the Marian toolkit in an attempt to improve translation quality.

3.1 Open-NMT

The OpenNMT system trained for this task used the the configuration for a large transformer network.

We used the following network hyperparameters:

- 1024 embedding size
- 4096 hidden units
- 12 layer encoder
- 12 layer decoder
- 16 transformer heads
- dropout 0.3
- attention dropout 0.1
- Tied embeddings for source, target and output layers
- Layer normalization
- Label smoothing
- Learning rate warm-up

The corpus was processed with SentencePiece using a model with a vocabulary size of 40K trained on the ru-en corpus. The network was trained for 10 epochs of this training data using a batch size of 1562, with an effective batch size of 24,992 using the lazy Adam (Kingma and Ba, 2015) optimizer. The final system was an average of the last 8 checkpoints of the training. Checkpoints were saved every 5000 steps. The system was then tuned with one epoch of newstest data from years 2014-2017.

3.2 Marian

Our Marian systems also utilize the transformer architecture. We use the WMT14 newstest2014 test set for validation during training and the following network hyperparameters:

- 2048 hidden units

[4] https://bitbucket.org/ndnlp/procrustes/src/master

- 6 layer encoder
- 6 layer decoder
- 8 transformer heads
- Tied embeddings for source, target and output layers
- Layer normalization
- Label smoothing
- Learning rate warm-up and cool-down

We first train a baseline system with the 58 million line corpus outlined in 2.1 and then train another baseline on the further-filtered 49 million line corpus outlined in 2.2. Using the word alignments generated earlier, we train systems utilizing alignments on subwords using fastalign (ga-spm-fastalign), alignments generated by projecting word-based fastalign alignments onto SentencePiece tokens (ga-procrusted-fastalign), and word-based MGIZA alignments onto SentencePiece tokens (ga-procrustes-mgiza). Results for decoding newstest2014 for each of these models are shown in Table 2.

system name	newstest2014
full-corpus baseline	39.81
ga-baseline	34.17
ga-spm-fastalign	33.06
ga-procrustes-fastalign	33.49
ga-procrustes-mgiza	31.92

Table 2: Experimental results for both baseline and guided-alignment systems decoding WMT14 testset measured in cased, detokenized BLEU.

We see that the best performing system is the one trained on the larger corpus, which is not surprising. We also see that while none of the guided-alignment based approaches we tried scored higher than the baseline on the smaller guided-alignment corpus, using the fastalign projected alignments performs better than the fastalign subword-based alignments by approximately 0.4 BLEU. We did experience issues getting MGIZA to successfully run on the 49 million line corpus, which may suggest additional processing of the training corpus is necessary to generate "correct" alignments using that approach. However, this specific MGIZA run provided the word alignments used in the Moses system described in the next section. This suggests more careful examination may be necessary before drawing conclusions as to the efficacy of using guided alignments to the Marian training process.

3.3 Moses

As in previous years, we trained a phrase-based Moses (Koehn et al., 2007) system with the guided-alignment data outlined in Section 2.2 in order to provide diversity for system combination. This system employed a hierarchical reordering model (Galley and Manning, 2008) and 5-gram operation sequence model (Durrani et al., 2011). The 5-gram English language model was trained with KenLM (Heafield, 2011) on the constrained monolingual corpus from our WMT15 (Gwinnup et al., 2015) efforts. System weights were tuned with the Drem (Erdmann and Gwinnup, 2015) optimizer using the "Expected Corpus BLEU" (ECB) metric.

3.4 System Combination

Once again, Jane (Freitag et al., 2014) system combination was used to combine various systems, tuned on newstest2016. We were able to successfully combine variations of three and four input systems, with results discussed in the following section.

4 Experimental Results

Results of decoding our various MT systems on WMT test sets from 2014 through 2019 are shown in Table 3.

Marian-base is an ensemble of 5 transformer models trained with identical hyperparameters as outlined in Section 3.2, with the exception of the initial random seed and using the language-id filtered corpus described in Section 2.1. Individual model weights are trained via Drem (Erdmann and Gwinnup, 2015) as outlined in last year's system.

Marian-ga is an ensemble of the four guided-alignment models described in Section 3.2: ga-baseline, ga-spm-fastalign, ga-procrustes-fastalign and ga-procrustes-mgiza. Individual model weights are also trained with Drem.

Onmt-base is the baseline system described in Section 3.1 and onmt-tune is the system that was further finetuned on newstest 2014-2017; Scores on those test sets are not reported due to overfitting during the fine tuning process.

Variations of system combinations are also reported - again with the absence of onmt-tune due to concerns of overfitting as newstest2016 is the test set used for tuning the system combination process. Combinations of only two systems resulted in a segmentation fault during processing due to fragility in the combination process.

We entered System 8 as our primary submission due to its performance gain on newstest2019 in a general (non-finetuned) setting, with the intuition that this years test set would discuss similar topics or issues as last years, while the earlier sets may be dated. In contrast, we submit System 5 as an alternative due to finetuning adapting the model to the collection of recent test sets.

5 Conclusion

In addition to our "known-good" approaches with increased data to submit respectably-performing translation systems, we conducted several experiments with guided alignments. Although these systems didn't outperform our prior approaches, they did figure into our final system combination submitted to the evaluation.

The authors wish to thank David Chiang for his implementation of the Procrustes alignment-projection script. The authors would also like to thank Grant Erdmann, Emily Conway and Grace Smith for their assistance in human evaluation of MT output.

References

2020. Findings of the 2020 conference on machine translation (WMT20). In *Proceedings of the Fifth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Tamer Alkhouli, Gabriel Bretschner, and Hermann Ney. 2018. On the alignment problem in multi-head attention-based neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 177–185, Brussels, Belgium. Association for Computational Linguistics.

Nadir Durrani, Helmut Schmid, and Alexander Fraser. 2011. A joint sequence translation model with integrated reordering. In *Proc. of the 49th Annual Meeting of the Association for Computational Linguistics (ACL '11)*, pages 1045–1054, Portland, Oregon.

Opinions, interpretations, conclusions and recommendations are those of the authors and are not necessarily endorsed by the United States Government. Cleared for public release on 1 October 2020. Originator reference number RH-20-121361. Case number 88ABW-2020-3049.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648, Atlanta, Georgia. Association for Computational Linguistics.

Grant Erdmann and Jeremy Gwinnup. 2015. Drem: The AFRL submission to the WMT15 tuning task. In *Proc. of the Tenth Workshop on Statistical Machine Translation*, pages 422–427, Lisbon, Portugal.

Miquel Esplà, Mikel Forcada, Gema Ramírez-Sánchez, and Hieu Hoang. 2019. ParaCrawl: Web-scale parallel corpora for the languages of the EU. In *Proceedings of Machine Translation Summit XVII Volume 2: Translator, Project and User Tracks*, pages 118–119, Dublin, Ireland. European Association for Machine Translation.

Markus Freitag, Matthias Huck, and Hermann Ney. 2014. Jane: Open source machine translation system combination. In *Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 29–32, Gothenburg, Sweden. Association for Computational Linguistics.

Michel Galley and Christopher D. Manning. 2008. A simple and effective hierarchical phrase reordering model. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 848–856.

Qin Gao and Stephan Vogel. 2008. Parallel implementations of word alignment tool. In *Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, pages 49–57, Columbus, Ohio. Association for Computational Linguistics.

Jeremy Gwinnup, Tim Anderson, Grant Erdmann, and Katherine Young. 2018. The AFRL WMT18 systems: Ensembling, continuation and combination. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 394–398. Association for Computational Linguistics.

Jeremy Gwinnup, Tim Anderson, Grant Erdmann, Katherine Young, Christina May, Michaeel Kazi, Elizabeth Salesky, and Brian Thompson. 2015. The AFRL-MITLL WMT15 system: There's more than one way to decode it! In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 112–119, Lisbon, Portugal. Association for Computational Linguistics.

Jeremy Gwinnup, Timothy Anderson, Grant Erdmann, Katherine Young, Michaeel Kazi, Elizabeth Salesky, Brian Thompson, and Jonathan Taylor. 2017. The AFRL-MITLL WMT17 systems: Old, new, borrowed, BLEU. In *Proceedings of the Second Conference on Machine Translation*, pages 303–309, Copenhagen, Denmark. Association for Computational Linguistics.

#	system name	WMT newstest 2014	2015	2016	2017	2018	2019
1	marian-base	41.09	35.30	35.64	38.89	34.03	37.04
2	marian-ga	32.86	27.97	28.46	31.49	27.43	31.96
3	moses-base	35.47	30.85	30.69	33.62	28.16	32.38
4	onmt-base	36.87	32.58	32.48	35.50	30.76	38.26
5	onmt-tune	–	–	–	–	32.31	39.27
6	syscomb 1+2+4	40.91	35.74	36.07	39.4	33.95	38.29
7	syscomb 1+3+4	41.00	35.85	35.87	39.52	33.99	38.24
8	syscomb 1+2+3+4	40.89	35.97	36.12	39.15	33.75	39.21

Table 3: Experimental results for both input systems and system combination results decoding WMT testsets measured in cased, detokenized BLEU as scored by mteval-13a.pl.

Kenneth Heafield. 2011. KenLM: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland. Association for Computational Linguistics.

Hieu Hoang, Tomasz Dwojak, Rihards Krislauks, Daniel Torregrosa, and Kenneth Heafield. 2018. Fast neural machine translation implementation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 116–121. Association for Computational Linguistics.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016a. Fasttext.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016b. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Guillaume Klein, Yoon Kim, Yuntian Deng, Vincent Nguyen, Jean Senellart, and Alexander Rush. 2018. OpenNMT: Neural machine translation toolkit. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Papers)*, pages 177–184, New Orleans. Association for Machine Translation in the Americas.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proc. of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 177–180.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán. 2019. Wikimatrix: Mining 135m parallel sentences in 1620 language pairs from wikipedia. *CoRR*, abs/1907.05791.

Rico Sennrich, Alexandra Birch, Anna Currey, Ulrich Germann, Barry Haddow, Kenneth Heafield, Antonio Valerio Miceli Barone, and Philip Williams. 2017. The University of Edinburgh's neural MT systems for WMT17. In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 389–399, Copenhagen, Denmark. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725. Association for Computational Linguistics.

Jason R. Smith, Herve Saint-Amand, Magdalena Plamada, Philipp Koehn, Chris Callison-Burch, and Adam Lopez. 2013. Dirt cheap web-scale parallel text from the common crawl. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL '13)*, pages 1374–1383, Sofia, Bulgaria.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 6000–6010.

Michał Ziemski, Marcin Junczys-Dowmunt, and Bruno Pouliquen. 2016. The united nations parallel corpus v1.0. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*.

The Ubiqus English-Inuktitut System for WMT20

François Hernandez
Ubiqus, Paris, France
fhernandez@ubiqus.com

Vincent Nguyen
Ubiqus, Paris, France
vnguyen@ubiqus.com

Abstract

This paper describes Ubiqus' submission to the WMT20 English-Inuktitut shared news translation task. Our main system, and only submission, is based on a multilingual approach, jointly training a Transformer model on several agglutinative languages. The English-Inuktitut translation task is challenging at every step, from data selection, preparation and tokenization to quality evaluation down the line. Difficulties emerge both because of the peculiarities of the Inuktitut language as well as the low-resource context.

1 Introduction

Ubiqus participated in the English to Inuktitut news translation task of WMT20. We performed a single submission, based on an unconstrained multilingual setup. The approach consists of jointly training a traditional Transformer (Vaswani et al., 2017) model on several agglutinative languages in order to benefit from them for the low-resource English-Inuktitut task (Aharoni et al., 2019).

Though the dataset provided for the task is sizable, with more than a million segments, it's quite narrow domain-wise, as it comes from proceedings of the Nunavut Hansard. The task being translation of news, it's expected to be a much wider domain. For that purpose, we extended the task with datasets of other - linguistically near - languages, as well as in-house datasets introducing more diversity to the domain.

All experiments were performed with the OpenNMT (Klein et al., 2017) toolkit, with *Tokenizer* [1] for data preprocessing and *OpenNMT-py* [2] for model training and inference.

[1]https://github.com/OpenNMT/Tokenizer
[2]https://github.com/OpenNMT/OpenNMT-py

2 Data

2.1 Training corpora

Based on prior internal work on English-Inuktitut translation tasks as well as other low-resource tasks, we focused our experiments on multilingual setups. Inuktitut is an agglutinative language, with a lot of particularities. Some *Uralic languages* like Finnish and Estonian can be considered close to Inuktitut in some linguistic aspects.

Most of our experiments are *unconstrained* with regards to the original WMT task in three ways:

- some datasets are taken from previous WMT tasks (English-Finnish, English-Estonian);
- some datasets are not in the WMT scope (more recent ParaCrawl[3] versions);
- some datasets were built in-house at Ubiqus Labs.

Some Inuktitut resources can easily be found on the internet, mostly from official government of Nunavut websites and initiatives. We performed two sets of data retrieval: a first one based on parallel crawling of multilingual websites, and a second one based on manual retrieval of parallel documents (mostly in PDF format) which then were automatically aligned with a commercial tool. In prior experiments, we also built a set of parallel news articles. Articles were manually retrieved and aligned from both the *Inuktitut* [4] magazine, which provides parallel versions of all its content in English, French, Inuktitut and Inuinnaqtun, and the Nunatsiaq News[5] website, which provides part of its content in both Inuktitut and English. We decided not to include this last dataset because of

[3]https://paracrawl.eu
[4]https://www.itk.ca/category/inuktitut-magazine/
[5]https://nunatsiaq.com

Proceedings of the 5th Conference on Machine Translation (WMT), pages 213–217

its high proximity with the *newsdev2020* and *newstest2020* of the task.

A summary of all the datasets used in the experiments is available in Table 1.

2.2 Evaluation sets

During our experiments, we conducted evaluation of the trained models with the provided *newsdev2020-eniu* as well as the *dev*, *devtest* and *test* parts of the Hansard dataset split. The latter were deduplicated prior to evaluation.

As a big part of our experiments revolve around multilingual aspects, we also used *newstest2018-enfi* and *newstest2019-enfi* for English-Finnish, as well as *newstest2018-enet* for English-Estonian.

Finally, we also conducted some evaluation over the test part of our in-house dataset built from *Inuktitut* magazine.

2.3 Data selection and cleaning

Deduplication as well as a few steps of cleaning were applied to every dataset. This consists of removing segments where:

- average token length too short or too long;
- source is strictly equal to target;
- numbers do not match between source and target side;
- source to target character ratio is too extreme.

The difference between raw and selected dataset size is shown in Table 1. It is noticeable that this step is especially important for our in-house datasets, where automatically crawled and aligned data is particularly messy. Also, it seems the Nunavut Hansard dataset is quite clean but contains a lot of duplicates.

2.4 Preprocessing and Tokenization

We decided to work on romanized Inuktitut. This allows straightforward parameter and vocabulary sharing in a basic bilingual English-Inuktitut setup, as well as maximizing the potential benefits of parameter sharing in a multilingual setup. Hence, all the Inuktitut data was romanized prior to any other processing, and we only converted back our *newstest2020-eniu* inferred hypothesis for submission.

All experiments were conducted on data tokenized with a BPE (Sennrich et al., 2016b) model with 12,000 merge operations, learned on the concatenation of all datasets – both source and target – presented in Table 1 (without any particular sampling strategy). This leads to a final vocabulary size of approximately 14k tokens. The choice of a smaller number of BPE merge operations stems from the agglutinative aspect of the language, leading us to think that dividing long tokens into more subwords might be beneficial to learn and share more useful representations. This seems to be also the approach in the baseline system proposed in (Joanis et al., 2020).

3 Experiments

3.1 Mixing languages

The method used to train models on multiple languages relies on the dataset weighting mechanism which is implemented within OpenNMT-py (Klein et al., 2020). When building batches, $weight_A$ examples are sampled from dataset A, then $weight_B$ from dataset B, and so on. This allows to dynamically subsample or oversample any specific dataset or language pair when training.

In order to allow Many-to-Many translation in a single shared model, we need to prepend each source with a tag indicating the target language (Johnson et al., 2017).

3.2 Bilingual only

Since we do not have any internal resource to assess the Inuktitut output, we started some bilingual experiments into English. With the English-Inuktitut datasets only, we realized that even with a base Transformer, the model converged very quickly and gave similar results with several varying hyper parameters. Also, changing the sampling weight of each sub-dataset did not have much impact to the final results. Moreover, English to Inuktitut bilingual experiments gave very poor results on our internal test set based on the Inuktitut Magazine. We hypothesize that there was some kind of overfitting to the Hansard domain. This is why we decided to extend a multilingual set up with more "news" based data.

3.3 Multilingual

We trained a few systems in the following order:

- first, a bilingual (and bidirectional) English-Inuktitut system (base configuration Transformer) using the Nunavut dataset as well as our in-house Web and Documents datasets;

Dataset	Origin	Raw	Selected
Nunavut Hansard v3.0 (Joanis et al., 2020)	WMT EN-IU 2020 Task	2,550,682	737,375
⋆Europarl English-Finnish	WMT EN-FI 2019 Task	1,969,624	1,564,994
⋆Europarl English-Estonian	WMT EN-ET 2018 Task	651,236	566,815
⋆ParaCrawlv6 English-Finnish	ParaCrawl Project	4,286,642	4,207,262
⋆ParaCrawlv6 English-Estonian	ParaCrawl Project	1,785,161	1,755,013
⋆Public Documents	Ubiqus	102,567	66,159
⋆Public Websites	Ubiqus	2,035,594	31,025

Table 1: Characteristics of the datasets used in the experiments. Datasets marked with ⋆ are considered out of the constraints of the WMT English-Inuktitut task.

- next, we added the English-Finnish data;
- then, we added the English-Estonian data;
- finally, we increased the model size.

Results for these systems are summed up in Table 3. We notice that multilingual setups are truly multilingual, in the sense that they provide output in the correct language, even though the scores are not very competitive (approx. 30% below the best scores at the time of the corresponding WMT tasks).

We decided to retain the bigger model (medium Transformer) for the submission. Bigger multilingual models tend to be better with regards to human evaluation, probably because the tasks are better spread across the parameters. This can be a problem in case of overfitting, which does not seem to be the case here as the scores remain in the same range. Also, the bigger model seems to give marginally better results in the additional tasks (Finnish and Estonian), which leads us to think it will be more robust to new test sets.

The configuration used for the final submission is the following:

- **Corpora and weights**: shown in Table 2.
- **Tokenization**: 12,000 BPE merge operations, learned on the concatenation of all datasets.
- **Model**: Transformer Medium (12 $heads$, $d_{model} = 768$, $d_{ff} = 3072$), with Relative Position Representations (Shaw et al., 2018).
- **Training**: Trained with OpenNMT-py on 6 RTX 2080 Ti, using mixed precision. Initial batch size is around 50,000 tokens, final batch size around 200,000 tokens. Training was stopped at 100k steps. Averaging was done continuously through exponential moving average.
- **Inference**: Shown scores are obtained with beam search of size 5 and average length penalty.

Hansard	15
⋆Europarl en-et	2
⋆Europarl en-fi	2
⋆ParaCrawlv6 en-et	10
⋆ParaCrawlv6 en-fi	10
⋆Public Documents (Ubiqus)	5
⋆Public Websites (Ubiqus)	1

Table 2: Dataset weighting used for the submitted system.

4 Future work

Our experiments remain in a rather traditional Neural Machine Translation scope, with the only addition of multiple languages and dataset weighting. Several paths can be explored from this starting point, such as adding more data for the current languages in the setup, authentic or synthetic (e.g. via back-translation (Sennrich et al., 2016a)), or adding other languages that might share some common characteristics, like Hungarian for instance.

Some additional work could also be explored on the tokenization part. For simplicity, our first approach in this paper relies on a very simple shared BPE approach. But, some more sophisticated approaches, maybe language-specific or morphologically adapted (Micher, 2018), may be worth exploring.

System	nd20-eniu	dev	dev-test	test	IM	nt18-enfi	nt19-enfi	nt18-enet
(Joanis et al., 2020)	-	24.2	17.9	19.3	-	-	-	-
en↔iu (base)	15.6	23.9	17.7	19.4	4.7	-	-	-
en↔iu/fi (base)	15.6	23.6	17.5	19.2	7.6	11.8	16.3	2.4
en↔iu/fi/et (base)	15.5	23.3	17.4	18.9	7.6	11.9	16.6	16.9
▷ en↔iu/fi/et (medium)	15.6	23.6	17.3	19.1	7.4	12.1	17.0	17.1

Table 3: BLEU (Papineni et al., 2002) scores for our various experiments, obtained with SacreBLEU (Post, 2018) v1.3.7. The submitted system is marked with ▷. *dev*, *dev-test* and *test* refer to the Hansard dataset evaluation sets. IM stands for *Inuktitut Magazine*.)

Finally, some more novel approaches could be tried, like massive pre-training methods such as BART (Lewis et al., 2019). A similar experimental process could be followed, starting from only the core languages of the task (English and Inuktitut), then extending to other languages and observe the impact.

5 Conclusion

Working on a new, unknown, language is always challenging. Even more so when this language is quite distant from any language you're used to. Also, automated metrics are far from being perfect for such tasks, especially in the context of such a particular language as Inuktitut.

Particularly for this task, human evaluation is key. But, as data, it's quite a scarce resource for Inuktitut. More knowledge of the language would be of tremendous help to better grasp the limits or interesting leads of the various models. One workaround can be to work on the opposite direction (Inuktitut to English), but there is no guarantee the model would have similar behaviour for similar tricks. And, some knowledge about Inuktitut would still be needed to analyze model behavior based on source inputs.

References

Roee Aharoni, Melvin Johnson, and Orhan Firat. 2019. Massively multilingual neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3874–3884, Minneapolis, Minnesota. Association for Computational Linguistics.

Eric Joanis, Rebecca Knowles, Roland Kuhn, Samuel Larkin, Patrick Littell, Chi kiu Lo, Darlene Stewart, and Jeffrey Micher. 2020. The Nunavut Hansard Inuktitut–English parallel corpus 3.0 with preliminary machine translation results. In *Proceedings of LREC-2020*, Marseille, France.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Guillaume Klein, François Hernandez, Vincent Nguyen, and Jean Senellart. 2020. The OpenNMT neural machine translation toolkit: 2020 edition. In *Proceedings of the 14th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pages 102–109, Virtual. Association for Machine Translation in the Americas.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Vancouver, Canada. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension.

Jeffrey Micher. 2018. Using the Nunavut hansard data for experiments in morphological analysis and machine translation. In *Proceedings of the Workshop on Computational Modeling of Polysynthetic Languages*, pages 65–72, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 464–468, New Orleans, Louisiana. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

SJTU-NICT's Supervised and Unsupervised Neural Machine Translation Systems for the WMT20 News Translation Task

Zuchao Li[1,2,3], **Hai Zhao**[1,2,3,*],
Rui Wang[4,*], **Kehai Chen**[4], **Masao Utiyama**[4], and **Eiichiro Sumita**[4]
[1]Department of Computer Science and Engineering, Shanghai Jiao Tong University (SJTU)
[2]Key Laboratory of Shanghai Education Commission for Intelligent Interaction
and Cognitive Engineering, Shanghai Jiao Tong University, Shanghai, China
[3]MoE Key Lab of Artificial Intelligence, AI Institude, Shanghai Jiao Tong University, China
[4]National Institute of Information and Communications Technology (NICT), Kyoto, Japan
charlee@sjtu.edu.cn, zhaohai@cs,sjtu.edu.cn, wangrui@nict.go.jp

Abstract

In this paper, we introduced our joint team SJTU-NICT 's participation in the WMT 2020 machine translation shared task. In this shared task, we participated in four translation directions of three language pairs: English-Chinese, English-Polish on supervised machine translation track, German-Upper Sorbian on low-resource and unsupervised machine translation tracks. Based on different conditions of language pairs, we have experimented with diverse neural machine translation (NMT) techniques: document-enhanced NMT, XLM pre-trained language model enhanced NMT, bidirectional translation as a pre-training, reference language based UNMT, data-dependent gaussian prior objective, and BT-BLEU collaborative filtering self-training. We also used the TF-IDF algorithm to filter the training set to obtain a domain more similar set with the test set for finetuning. In our submissions, the primary systems won the first place on English to Chinese, Polish to English, and German to Upper Sorbian translation directions.

1 Introduction

Our SJTU-NICT team participated in the WMT20 shared task, including supervised track, unsupervised, and low-resource track. During the participation, we placed our attention on Polish (PL) → English (EN) and English (EN) → Chinese (ZH) on the supervised track, while on the unsupervised and low-resource track, the German (DE) ↔ Upper Sorbian (HSB) both directions are focused.

Our baseline system in supervised track is based on the Transformer big architecture proposed by Vaswani et al. (2017), in which its open-source implementation version `Fairseq` (Ott et al., 2019) is adopted. In the unsupervised and low-resource track, we draw on the successful experience of the `XLM` framework (Conneau et al., 2019), and used the two-stage training mode of masked language modeling (MLM) pre-training + back-translation (BT) finetune to obtain a very strong baseline performance. `Marian` (Junczys-Dowmunt et al., 2018) toolkit is utilized for training the decoder in reranking using machine translation targets instead of common GPT-style language modeling targets.

In order to better play the role of WMT evaluation in polishing the methods proposed or improved by our team (He et al., 2018; Li et al., 2018; Zhang et al., 2018; Zhang and Zhao, 2018; Xiao et al., 2019; Zhou and Zhao, 2019; Li et al., 2019b; Luo and Zhao, 2020), we divided the three language pairs we participated in into three categories:

1. Traditional language pair with rich parallel corpus: EN-PL,
2. Language pair with document-level information: EN-ZH,
3. Language pair with no or low parallel resources: DE-HSB.

In the supervised PL→EN translation direction, we based on the XLM framework to pre-train a Polish language model using common crawl and news crawl monolingual data, and proposed the XLM enhanced NMT model inspired from the idea of incorporating BERT into NMT (Zhu et al., 2020). Besides, we trained a bidirectional translation model of EN-PL based on the parallel corpus and further finetuned it to the PL→EN

* Corresponding authors. This paper was partially supported by National Key Research and Development Program of China (No. 2017YFB0304100), Key Projects of National Natural Science Foundation of China (U1836222 and 61733011), Huawei-SJTU Long Term AI Project, Cutting-edge Machine Reading Comprehension and Language Model. Rui Wang was partially supported by JSPS grant-in-aid for early-career scientists (19K20354): "Unsupervised Neural Machine Translation in Universal Scenarios" and NICT tenure-track researcher startup fund "Toward Intelligent Machine Translation".

Proceedings of the 5th Conference on Machine Translation (WMT), pages 218–229
Online, November 19–20, 2020.

direction.

In the supervised EN→ZH translation with document information, we propose a document enhanced NMT model based on Longformer (Beltagy et al., 2020). The training of our proposed document enhanced NMT model is split into three stages. In the first stage, we pre-train the Longformer document encoder with MLM target on the document text in Wikipedia dumps, UN News, and News Commentary monolingual corpus. A conventional Transformer-big NMT model is trained in the second stage. In the final stage, the Longformer encoder and conventional Transformer big NMT model are used to initialize the full document-enhanced NMT model parameters, in which the Longformer encoder is adopted to extract representations for the document of an input sequence, and then the document representations are fused with each layer of the encoder and decoder of the NMT model through attention mechanisms.

In the unsupervised machine translation track on DE-HSB, we experimented with the reference language based UNMT (RUNMT) (Li et al., 2020b) framework we proposed recently. Under this framework, we choose English as the reference language, and use the Europarl parallel corpus of EN-DE to enhance the unsupervised machine translation between DE and HSB. Specifically, we adopted reference language translation (RAT), reference language back-translation (RABT), and cross-lingual back-translation (XBT) three training targets with the help of the cross-lingual agreement provided by the EN-DE parallel corpus to enhance the unsupervised translation performance.

Due to the introduction of more explicit supervision signals brought by parallel corpus in the low-resource machine translation track on DE-HSB, we discarded the use of the weaker agreement provided by the reference language, conducted joint training on the unsupervised back-translation and the supervised (forward-)translation directly, and introduced BT-BLEU based collaborative filtering technology for further self-training. In addition, inspired by our previous work (Sun et al., 2020b), we also use MLM and translation language modeling (TLM) to continue pre-training the model while machine translation training.

In addition, in all basic NMT models, we empower the training process with our proposed data-dependent gaussian prior objective (D2GPo) (Li et al., 2020a), so that the model can maintain the diversity of the output. When the main model training is finished, the TF-IDF algorithm is employed to filter the training set according to the input of the test set, a training subset whose domain is more similar to the test set is obtained, and then used to finetune the model for reducing the performance degradation caused by domain inconsistency. For the final submission, an ensemble of several different trained models outputs the n-best predictions, and used the decoder trained with `Marian` toolkit to performs reranking to get the final system output.

2 Methodology

2.1 XLM-enhanced NMT

Pre-trained language models such as ELMo (Peters et al., 2018), BERT (Devlin et al., 2019), XLM (Conneau et al., 2019), XLNet (Yang et al., 2019), ALBERT (Lan et al., 2019) etc. have recently demonstrated a very dominant effect on natural language processing tasks. Several works (Clinchant et al., 2019; Imamura and Sumita, 2019; Zhu et al., 2020) leveraged a pre-trained BERT model for improving NMT and found that BERT can bring significantly better results over the baseline.

Since BERT and other pre-trained language models are trained on large scale corpus beyond the data provided by the WMT20 organizers, the direct use of BERT will make the system submitted unconstrained. Using an XLM model, a variant of BERT, pre-trained from scratch on the monolingual data provided by the official to enhance our NMT model, is a good choice to keep the system constrained. Moreover, the XLM model has the advantages of simple training preprocessing, low requirement for training environment that no specialized hardware such as TPU is needed. Inspired by the *BERT-fused model* proposed by Zhu et al. (2020), we built a *XLM-enhanced model*, in which we utilize XLM context-aware representations to adaptively interact with all layers in the NMT model with attention mechanism, instead of serving it as input embeddings only.

In the *XLM-enhanced model*, XLM as an additional encoder and the original encoder of NMT constitute a dual-encoder structure, which is very similar to our previous work (Li et al., 2019a). The XLM-encoder attention and XLM-decoder attention are essentially the same with the

Representation Learning Frameworks (RLFs) we proposed: Source-side fusion RLF (SRLF), Target-side fusion RLF (TRLF), and both-side fusion RLF (BRLF, which is a combination of SRLF and TRLF). Specifically, in the SRLF, given a source language input x, a Pre-trained Language Modeling (PLM) encoder (like BERT, XLM) first encodes it into a context-aware representation:

$$H_P = \texttt{PLM}^k(x), \tag{1}$$

where H_P is the output of the k-th layer of the PLM encoder. As PLM and NMT models adopt different sub-word segmentation rules or algorithms and the addition of special tokens are different, the input sequence length of PLM and NMT encoders is inconsistent or cannot correspond in every position. Assuming that i represents the position of the input sequence of NMT encoder, the hidden state H_E^l after fusion with H_P in SRLF of the l-th layer is:

$$\begin{aligned} H_E^l = \frac{1}{2}(&\texttt{attn}_\texttt{S}(H_E^{l-1}, H_E^{l-1}, H_E^{l-1}) \\ &+\texttt{attn}_\texttt{P}(H_E^{l-1}, H_P, H_P)), \end{aligned} \tag{2}$$

where $\texttt{attn}_S$ is a multi-head self-attention layer and $\texttt{attn}_P$ is the multi-head attention layer. H_E will eventually be output from the last layer as the final representation.

In the TRLF framework, the dual-encoder provides two encoded outputs; the decoder will use both contexts at the same time. In the case of layer l in the decoder, we have

$$\begin{aligned} H_{DS}^l &= \texttt{attn}_\texttt{MS}(H_D^{l-1}, H_D^{l-1}, H_D^{l-1}), \\ H_D^l &= \frac{1}{2}(\texttt{attn}_\texttt{EC}(H_{DS}^l, H_E, H_E) \\ &\quad +\texttt{attn}_\texttt{PC}(H_{DS}^l, H_P, H_P)), \end{aligned} \tag{3}$$

where $\texttt{attn}_\texttt{MS}$ is the multi-head future-masked self-attention layer, $\texttt{attn}_\texttt{EC}$ and $\texttt{attn}_\texttt{PC}$ are independent multi-head attention layer for context query.

In the condition that SRLF framework is only used, the representation of PLM is only fused into the final representation H_E in the encoder side; then the decoder side continues to use the original decoding ways: $H_D^l = \texttt{attn}_\texttt{PC}(\texttt{H}_\texttt{DS}^\texttt{l}, \texttt{H}_\texttt{E}, \texttt{H}_\texttt{E})$. While the the TRLF framework is only adopted, the output of NMT encoder is $H_E = \texttt{attn}_\texttt{S}(\texttt{H}_\texttt{E}^{\texttt{l}-1}, \texttt{H}_\texttt{E}^{\texttt{l}-1}, \texttt{H}_\texttt{E}^{\texttt{l}-1})$. A BRLF framework is a combination of these two frameworks.

Moreover, in the training of the RLFs, a same drop-net trick proposed by Zhu et al. (2020) is adopted to ensure that the features output by PLM and the conventional encoder are fully utilized. In this method, the interval of 0-1 is divided into three parts according to the pre-set drop-net ratio p_{net}, where $[0, \frac{p_{net}}{2})$ is the probability of attending to the final sum for the first `attn` in H_E^L and H_D^L, $[\frac{p_{net}}{2}, 1 - \frac{p_{net}}{2})$ is the probability for the whole H_E^L and H_D^L equation, $[1 - \frac{p_{net}}{2}, 1]$ is the probability for the second `attn` in in H_E^L and H_D^L.

2.2 Bidirectional NMT

Machine translation, in general, is unidirectional, that is, from the source language to the target language. The encoder-decoder framework for NMT has been shown effective in large data scenarios, and the more high-quality bilingual training data, the better performance the model tends to achieve. Recent works (Zoph et al., 2016; Kim et al., 2019) on translation transfer learning (Torrey and Shavlik, 2010; Pan and Yang, 2009) from rich-resource language pairs to low-resource language pairs demonstrate that translation has some universal nature in essence between different language pairs. As the source-to-target (S2T) forward translation and target-to-source (T2S) backward translation can be seen as two special language pairs in bilingual translation, it can make use of the translation universal nature to improve each other, i.e., dual learning (He et al., 2016). Based on this motivation, we developed a bidirectional NMT model, in which the S2T and T2S translation were trained and optimized jointly. Therefore, the training data was doubled to make better and full use of the costly bilingual corpus.

Given parallel corpus $\mathcal{C} = \{(x^{(n)}, y^{(n)})\}_{n=1}^N$, the bidirectional NMT model is trained in two phase. In the first *bidirectional translation as pre-training* phase, a joint training objective is used to jointly maximize the likelihood of both translation direction on the bilingual data:

$$\mathcal{L}(\theta_{parent}) = \sum_{n=1}^{N}(\log p(y^{(n)}|x^{(n)}) + \log p(x^{(n)}|y^{(n)})), \tag{4}$$

where θ_{parent} is the parameters of the model, namely *parent model*, obtained in this phase.

The second phase is *unidirectional translation fine-tuning*. Although there are commonalities in different translation directions, the differences are also very obvious. To further expose the model to the direction difference and improve the effect of unidirectional translation, we further finetune the

bidirectional pre-trained model on the bilingual data. Take S2T translation as an example; the model is optimized as follows:

$$\mathcal{L}(\theta_{S\to T}) = \sum_{n=1}^{N} \log p(y^{(n)}|x^{(n)}), \qquad (5)$$

where $\theta_{S\to T}$ is the parameters of *child model* which is initialized with θ_{parent}. Similarly, the T2S *child model* can also be obtained.

Due to the introduction of bidirectional translation in one model, follow the practice of Conneau and Lample (2019), shared subword vocabulary and shared encoder-decoder (source and target) embedding were employed to improves the alignment of embedding spaces across languages. In addition, since the encoder and decoder need to be able to handle two languages simultaneously, a language embedding was used to indicate the language being processed, so as to reduce confusion of the model.

2.3 Document-enhanced NMT

In spite of its success (Vaswani et al., 2017), sentence-level NMT has been based on strong independence and locality assumptions generally, in which the interrelations among these discourse (Jurafsky, 2000) elements were ignored. This results in that the translations may be perfect at the sentence-level but lack crucial properties of the text, hindering understanding (Maruf et al., 2019). To help to resolve ambiguities and inconsistencies in translations, some MT pioneers (Bar-Hillel, 1960; Xiong et al., 2013; Sennrich, 2018) exploit the underlying discourse structure information of a text to address this issue, while others (Bawden et al., 2018; Voita et al., 2018; Jean and Cho, 2019; Wang et al., 2019; Scherrer et al., 2019) extend the translation units with the context or use an additional context encoder and attention. It is worth noting that the essence of the document-level NMT claimed with additional context and attention is still sentence-level MT, whose translation is still output sentence by sentence. We named it as document-enhanced NMT more precisely.

Due to computational efficiency and tractability concerns, the document-enhanced NMT models mostly used document embedding, document topic information, and limited past or future context sentences, etc., rather than the truly whole document information. Recently, with the increase in computational power available to us and the well-designed neural network structures (Dai et al., 2019; Kitaev et al., 2019; Beltagy et al., 2020) for long sequence encoding, we are finally in a position to employ the whole document information for enhancing sentence-level NMT. In addition, we argue that since long sequences encoding is easier than decoding, truly whole document-level translation is still a long way off, since the bidirectional context is available in the encoder, but only the past is visible by the decoder.

Longformer To make the long documents processed with Transformer (Vaswani et al., 2017) architecture feasible or easier, a modified Transformer architecture named Longformer was proposed by Beltagy et al. (2020), in which the limitation for memory and computational requirements is addressed with a novel self-attention operation scales linearly with the sequence length.

In Longformer, the original full self-attention ($O(n^2)$ time and memory complexity) is sparsified to makes it efficient for longer sequences. There are three "attention patterns" for specifying pairs of input locations attending to one another.

- **Sliding Window** Self-attention is performed in a fixed-size window w and multiple stacked layers of such sliding windowed attention results in a large receptive field as analogs to CNNs.
- **Dilated Sliding Window** Inspired by the dilated CNNs (Oord et al., 2016), dilation gaps of size d is introduced to the window to further increase the receptive field without increasing computation.
- **Global Attention** Though the receptive field is enlarged by stacking multiple layers and dilation in sliding window and dilated sliding window attention patterns, some part of the long sequence has the requirement for keeping the full and global receptive field due to the downstream tasks, so global attention is introduced to make up this need.

In our document-enhanced NMT model, some heads in multi-head attention are set to use the sliding window pattern to focus on the local context which was revealed very important (Kovaleva et al., 2019), while others with dilation focus on longer context. Besides, as Longformer is incorporated into the NMT model, we perform

global attention on the position of `[CLS]` token in which the representation of the whole sequence (i.e., the document embedding) is generated. This makes the previous document-enhanced model with document embedding as a special case of ours. It is worth noting that since the sentence being translated is part of the document, setting its positions in the document to use global attention pattern will improve the performance; but to reduce the document computation and use cache for acceleration (not recalculate the document for each sentence), we only attend the `[CLS]` position globally.

In our document-enhanced model, the Longformer is first pre-trained with the masked language modeling objective on the monolingual document corpus. It is fixed throughout the NMT training to reduce the model parameters optimized in the training stage. Thus, Longformer can also be thought of as a pre-trained language model, as it provides a document context representation H_P for the NMT model, the integration of Longformer in *Document-enhanced NMT* is consistent with the XLM model in *XLM-enhanced NMT*.

2.4 Reference Language based UNMT

The rise of UNMT almost completely relieves the parallel corpus curse, though UNMT is still subject to unsatisfactory performance due to the vagueness of the clues available for its core back-translation training. Further enriching the idea of pivot translation by extending the use of parallel corpora beyond the source-target paradigm, we propose a new reference language-based framework for UNMT, RUNMT, in which the reference language only shares a parallel corpus with the source, but this corpus still indicates a signal clear enough to help the reconstruction training of UNMT through a proposed reference agreement mechanism.

Specifically, we proposed three kinds of reference agreement utilization approaches in (Li et al., 2020b): reference agreement translation (RAT), reference agreement back-translation (RABT), and cross-lingual back-translation (XBT).

RAT RAT utilizes the principle for translating paired sentences into the target language $\mathcal{T}$ of the source $\mathcal{S}$ and reference $\mathcal{R}$ language. Since the input the parallel, the both translation outputs should be the same. Given a parallel sentence pair $\langle s, r\rangle$ between language $\mathcal{S}$ and $\mathcal{R}$, we would ideally have $\mathbb{P}(\cdot|s;\theta_{\mathcal{S}\to\mathcal{T}}) = \mathbb{P}(\cdot|r;\theta_{\mathcal{R}\to\mathcal{T}})$, where $\theta_{\mathcal{S}\to\mathcal{T}}$ and $\theta_{\mathcal{R}\to\mathcal{T}}$ represent $\mathcal{S}\to\mathcal{T}$ and $\mathcal{R}\to\mathcal{T}$ translation models respectively. However, as the two models are trained on different data, the agreement may be corrupted. Therefore, we combine the two models to obtain the agreed-upon translation output $\tilde{t}_a$:

$$\tilde{t}_a \sim \mathbb{P}(\cdot|s,r;\theta_{\mathcal{S}\to\mathcal{T}},\theta_{\mathcal{R}\to\mathcal{T}}), \tag{6}$$

where $\mathbb{P}(\cdot|s,r;\theta_{\mathcal{S}\to\mathcal{T}},\theta_{\mathcal{R}\to\mathcal{T}})$ is

$$\prod_{i=1}^{J}[\frac{1}{2}(\mathbb{P}(\cdot|s,\tilde{t}_{<i};\theta_{\mathcal{S}\to\mathcal{T}}) + \mathbb{P}(\cdot|r,\tilde{t}_{<i};\theta_{\mathcal{R}\to\mathcal{T}}))], \tag{7}$$

$\tilde{t}_{<i}$ indicates the decoded tokens before the i-the generation step.

Finally, two synthetic sentence pairs $\langle s,\tilde{t}_a\rangle$ and $\langle r,\tilde{t}_a\rangle$ are used to train the models $\mathcal{S}\to\mathcal{T}$ and $\mathcal{R}\to\mathcal{T}$. Since the silver learning target is optimized, the smoothed cross-entropy loss $\mathcal{L}_\epsilon$ is used instead of the ordinary cross-entropy loss $\mathcal{L}$. The learning objective for RAT can be written as:

$$\mathcal{L}_{\mathbf{RAT}}(\mathcal{S},\mathcal{T},\mathcal{R}) = \mathcal{L}_\epsilon(\theta_{\mathcal{S}\to\mathcal{T}}) + \mathcal{L}_\epsilon(\theta_{\mathcal{R}\to\mathcal{T}}), \tag{8}$$

RABT With the regularized pseudo-parallel sentences in RAT, we not only train the $\mathcal{S}\to\mathcal{T}$ and $\mathcal{R}\to\mathcal{T}$ forward-translation models (as the generation direction is the same as the training direction), but also train the BT models, i.e., $\mathcal{T}\to\mathcal{S}$ and $\mathcal{T}\to\mathcal{R}$. The learning objective of RABT can be described as:

$$\mathcal{L}_{\mathbf{RABT}}(\mathcal{S},\mathcal{T},\mathcal{R}) = \mathcal{L}(\theta_{\mathcal{T}\to\mathcal{S}}) + \mathcal{L}(\theta_{\mathcal{T}\to\mathcal{R}}). \tag{9}$$

XBT The parallel corpus between languages $\mathcal{S}$ and $\mathcal{R}$ can not only bring agreement in the translations of the same target language $\mathcal{T}$, but also cross-lingual agreement, that is, using the target language as the bridge to form pivot translation (Wu and Wang, 2007; Utiyama and Isahara, 2007; Paul et al., 2009) patterns: $\mathcal{S}\to\mathcal{T}\to\mathcal{R}$ and $\mathcal{R}\to\mathcal{T}\to\mathcal{S}$. In XBT, paired sentences s and r are translated to language $\mathcal{T}$: $\tilde{t}_s$ and $\tilde{t}_r$, and forms two new pseudo-parallel pairs: $\langle\tilde{t}_s,r\rangle$ and $\langle\tilde{t}_r,s\rangle$, which promote the training of translation $\mathcal{T}\to\mathcal{R}$ and $\mathcal{T}\to\mathcal{S}$. The objective function of XBT is:

$$\mathcal{L}_{\mathbf{XBT}}(\mathcal{S},\mathcal{T},\mathcal{R}) = \mathcal{L}(\theta_{\mathcal{T}\to\mathcal{R}}) + \mathcal{L}(\theta_{\mathcal{T}\to\mathcal{S}}), \tag{10}$$

2.5 CFST: Collaborative Filter for Self-Training with BT-BLEU

Self-training, proposed by Scudder (1965), is a semi-supervised approach that utilizes unannotated

Algorithm 1 Classic Self-training

1: Train a base NMT/UNMT model $f_{\theta_{S\to T}}$ on $\mathcal{C}$
2: **repeat**
3: Apply $f_{\theta_{S\to T}}$ to the unlabeled instances $\mathcal{U}$
4: Select a subset $\mathcal{Q} \subset \{(x, f_{\theta_{S\to T}}(x))|x \in \mathcal{U}\}$
5: Update model $f_{\theta_{S\to T}}$ on $\mathcal{Q}$ with self-training objective and $\mathcal{C}$ with original objective
6: **until** convergence or maximum iterations are reached

data to create better models. Recently, self-training has been successfully applied to both NMT and UNMT fields (He et al., 2019; Sun et al., 2020a), especially for the unbalanced low-resource training data scenarios.

Formally, in self-training strategy for machine translation, a parallel dataset $\mathcal{C} = \{(x^{(n)}, y^{(n)})\}_{n=1}^{N}$ in NMT and a unpaired monolingual dataset $\mathcal{D} = \{x^{(m)}\}_{m=1}^{M} \cup \{y^{(n)}\}_{n=1}^{N}$ in UNMT is used to train the initial model. Then, a subset of pseudo parallel data is incorporated to update the model with a pseudo-supervised NMT (PNMT) objective (including forward translation and backward translation) for both NMT and UNMT as shown in Algorithm 1. In NMT, a large unlabeled dataset $\mathcal{U} = \{x^{(j)}\}_{j=1}^{L}$ is used for the synthesis of pseudo-parallel corpora. While in UNMT, since the model is trained with back-translation on unpaired monolingual data, the pseudo-parallel corpora is synthesized by the monolingual data, i.e., $\mathcal{U} = \{x^{(m)}\}_{m=1}^{M}$.

Considering the translation quality can't effectively be evaluated across languages in machine translation with only the monolingual data, therefore the selection of the subset $\mathcal{Q}$, is one of the key factors for self-training. It is usually selected based on some confidence scores (e.g. log probability or perplexity, PPL) (Yarowsky, 1995), but it is also possible for S to be the whole pseudo parallel data (Zhu and Goldberg, 2009). In the backward translation based on the pseudo-parallel data, the DAE method widely used in UNMT can alleviate the impact of the noise resulted from the synthesized sentences on model training, since the synthesized sentences are only used as input. However, in the forward translation training, the quality of noisy targets will directly affect the success of the model training. Therefore, the selection of synthetic parallel corpus becomes particularly critical.

Algorithm 2 BT-BLEU based Collaborative Filter

1: Split $\mathcal{U}$ equally into two subsets $\mathcal{U}_1 = \{x^{(j)}\}_{j=1}^{L/2}$ and $\mathcal{U}_2 = \{x^{(j)}\}_{j=L/2+1}^{L}$
2: Apply $f_{\theta_{S\to T}}$ to the unlabeled instances $\mathcal{U}_1$ and $\mathcal{U}_2$
3: Train two backward translation models $f^{(1)}_{\theta_{T\to S}}$ with $\{(f_{\theta_{S\to T}}(x), x)|x \in \mathcal{U}_1\}$ and $f^{(2)}_{\theta_{T\to S}}$ with $\{(f_{\theta_{S\to T}}(x), x)|x \in \mathcal{U}_2\}$ respectively
4: Translate $\{f_{\theta_{S\to T}}(x)|x \in \mathcal{U}_2\}$ with model $f^{(1)}_{\theta_{T\to S}}$, while $\{f_{\theta_{S\to T}}(x)|x \in \mathcal{U}_1\}$ with model $f^{(2)}_{\theta_{T\to S}}$
5: Calculate BT-BLEU $\mathcal{B}$ for two subsets: $\textbf{BLEU}(f^{(2)}_{\theta_{T\to S}}(f_{\theta_{S\to T}}(x)), x), \forall x \in \mathcal{U}_1$ and $\textbf{BLEU}(f^{(1)}_{\theta_{T\to S}}(f_{\theta_{S\to T}}(x)), x), \forall x \in \mathcal{U}_2$
6: $\mathcal{Q} = \{(x, f_{\theta_{S\to T}}(x))|x \in \mathcal{U}_1, \mathcal{B} > \gamma\} \cup \{(x, f_{\theta_{S\to T}}(x))|x \in \mathcal{U}_2, \mathcal{B} > \gamma\}$

We propose a collaborative filtering algorithm based on BT-BLEU to select high quality pseudo-parallel pairs, as shown in Algorithm 2. The BT-BLEU, as defined in (Li et al., 2020b), is a BLEU of $\boldsymbol{x} \in \mathcal{S}$ and $\tilde{\boldsymbol{x}}$ generated in the $\mathcal{S} \to \mathcal{T} \to \mathcal{S}$ back-translation process. As long as the model of $\mathcal{T} \to \mathcal{S}$ is fixed and the preference for translation of certain sentences is reduced as much as possible, BT-BLEU can reflect the translation quality of $\mathcal{S} \to \mathcal{T}$ to some extent, because of the necessary but insufficient condition that only the better the translation of $\mathcal{S} \to \mathcal{T}$ is, the better the translation of $\mathcal{T} \to \mathcal{S}$ can be.

To achieve the goal of reducing translation preferences, we split the pseudo parallel set into two subsets, ensure no overlap between two subsets. The model trained on subset 1 is used for back-translation on the subset 2, while the model on subset 2 back-translate the subset 1. This collaborative translation process enables the two models not to see the sentences to be translated, which guarantees the translation not relies on tricks. Additionally, we found that the sentences in different lengths have different difficulties for back-translation; we further divide the sentences into different bags according to their lengths and use different BT-BLEU threshold γ for filtering.

2.6 TF-IDF Finetune

NMT has been prominent in many machine translation tasks. However, in some domain-specific tasks, only the corpora from similar

Systems	Dev	Test	
	BLEU	BLEU	chrF
Base Data:			
Transformer big	25.8	-	-
XLM-enhanced	26.8	-	-
Base Data + ParaCrawl:			
Transformer big	30.0	32.2	0.596
+D2GPo	30.9	-	-
XLM-enhanced	31.4	-	-
Bidirectional NMT	29.5	-	-
+Finetune	31.2	-	-
Ensemble	32.0	34.0	0.606
++TF-IDF finetune	32.3	34.2	0.609
++Re-ranking	32.5	34.6	0.610

Table 1: PL→EN performance (sacreBLEU and chrF score) for different models.

domains can improve translation performance. If a trained NMT model is evaluated on a domain mismatch corpus, the translation performance may even degrade. Therefore, domain adaptation techniques are essential to solve the NMT domain problem. It is a very common domain adaptation approach to further finetune the translation model trained on the domain-mixed corpus by using data that is the same or similar to the test set in domain. Therefore, we need to select sentences that are as close to the input domain as possible in the domain-mixed training set.

We argue that low-frequency words contain more domain information than high-frequency words, since low-frequency words are mostly domain-specific nouns, etc., which may indicate the topic directly. Therefore, we adopt the TF-IDF algorithm to search and filter on the whole training set. In fact, the improved version of TF-IDF algorithm, BM25 (Robertson and Zaragoza, 2009), is employed to calculate the sentence similarity. BM25 is based on probabilistic information retrieval theory, whose score for a term q to a sequence Q is:

$$s(Q,q) = \frac{\text{IDF} * ((k+1) * \text{TF})}{(k * (1.0 - b + b * \frac{L_Q}{L_{\text{avg}}}) + \text{TF})}, \quad (11)$$

where IDF is the Inverse Document Frequency for term q appears in the whole corpus, TF is the Term Frequency for q in D, L_Q represents the sequence length, L_{avg} is the average length of corpus D, k and b is the adjustable parameters.

With this scorer, every sequence will obtain a BM25 vector on the terms of the corpus:

$$V = [s(Q,t), \quad \forall t \in D_{\text{terms}}], \quad (12)$$

where D_{terms} indicates the all terms set in corpus D. We calculate the cosine similarity as final scores between the query and every source sentence in corpus, and ranked on the scores to get the top-K pairs (K=1000 in our experiments) as the sub-training set for finetuning.

3 Data Preprocessing and Model Setup

Before model training, we preprocessed the data uniformly and customized the processing according to the requirements of each model. We normalized punctuation, remove non-printing characters, and tokenize all data with the Moses tokenizer (Koehn et al., 2007) except for the Chinese. For Chinese, we removed the segmentation space in some training data and then use PKUSeg (Luo et al., 2019) toolkit to cut all Chinese sentences, so as to obtain unified word segmentation annotations. We use joint byte pair encodings (BPE) with 40K split operations for subword segmentation (Sennrich et al., 2016).

In *XLM-enhanced NMT* and *Document-enhanced NMT*, we first train a basic NMT (Transformer big) model on the sentence-level data until convergence, then initialize the encoder and decoder of the *XLM-enhanced NMT* and *Document-enhanced NMT* full model with the obtained model. The PLM-encoder attention $\texttt{attn}_{\texttt{P}}$ and PLM-decoder attention $\texttt{attn}_{\texttt{PC}}$ are randomly initialized.

EN-PL On the language pair EN-PL, we explored performance in two training data settings. The first is *base data*, including Europarl v10, Tilde Rapid corpus, and WikiMatrix bitext data, whose raw data is on the sentence-level. In the second setting *base data + paracrawl*, we converted the paragraph-level alignment data in Paracrawl to sentence-level alignment and incorporated it with the *base data*. In the conversion process, we adopted the method and program proposed by (Gale and Church, 1993) for aligning sentences based on a simple statistical model of character lengths, which uses the fact that longer sentences in one language tend to be translated into longer sentences in the other language, and that shorter sentences tend to be translated into shorter

Systems	19test	Test	
	BLEU	BLEU	chrF
Transformer big	37.2	-	-
+D2GPo	37.7	-	-
XLM-enhanced	38.9	-	-
Document-enhanced	39.2	-	-
Ensemble	40.0	48.6	0.418
++TF-IDF finetune	40.2	48.8	0.422
++Re-ranking	40.5	49.1	0.427

Table 2: EN→ZH performance (charBLEU and chrF score) for different models.

sentences. A probabilistic score is assigned to each proposed correspondence of sentences, based on the scaled difference of lengths of the two sentences (in characters) and the variance of this difference. This probabilistic score is used in a dynamic programming framework to find the maximum likelihood alignment of sentences.

For the Polish pre-trained XLM language model, we used all NewsCrawl monolingual data and some CommonCrawl monolingual data. Since the CommonCrawl data is very large and noisy and can potentially decrease the performance of LM if it is used in its raw form. We apply language identification filtering (`langid`; Lui and Baldwin (2012)), keeping sentences with correct languages. In order to filter out the sentences shorter than 5 words or longer than 150 words more precisely, we re-split sentences using Spacy (Honnibal and Montani, 2017) toolkit.

EN-ZH In EN-ZH, the pre-training of Longformer as a document encoder is unique. As described in (Beltagy et al., 2020), the Longformer needs a large number of gradient updates to learn the local context first; before learning to utilize longer context. In the first phase of the staged training procedure, an initial RoBERTa (Liu et al., 2019) model implemented in Fairseq (Ott et al., 2019) repository was trained on the sentence-level text available. In each subsequent phase, we trained the model on the paragraph text, doubled the window size and the sequence length, and halve the learning rate. For the paragraph text, the Wikidumps and NewsCommentary v15 have document intervals and can be used directly, while UN v1.0 has no document intervals but the sentence order is not interrupted. Therefore, we use the BERT Next Sentence Prediction (NSP) classification model provided by Google for document interval prediction to recover the documents.

DE-HSB In RUNMT on EN-DE-HSB, Europarl v10 EN-DE parallel corpus is used for EN-DE NMT and RAT/RABT/XBT training[1]. Additionally, the BPE size increases to 50K for three languages. In CFST, the filtering threshold of BT-BTBLEU is set to $\gamma = 50.0$.

4 Results and Analysis

Results and ablations for PL→EN[2] are shown in Table 1, EN→ZH in Table 2, unsupervised DE↔HSB in Table 3 and low-resource DE↔HSB in Table 4. We report case-sensitive SacreBLEU scores using SacreBLEU (Post, 2018) for EN-PL, DE-HSB, and BLEU based on characters for EN-ZH. In the results, "+" means addition based on baseline, and "++" means cumulative addition based on the previous one.

In PL→EN, the introduction of ParaCrawl data improves the baseline performance on the dev dataset by about 4.2 BLEU. +D2GPo, XLM-enhanced NMT, Bidirectional NMT, and ensembling outperforms our strong baseline by 2 BLEU point. Finally, finetuning and reranking further gives another 0.5 BLEU.

For EN→ZH, as with PL→EN, we see similar improvements with +D2GPo, XLM-enhanced NMT, ensembling and reranking. We also observe that the addition of Document-enhanced NMT is much more substantial, improving single model performance by over 1.5 BLEU.

In the unsupervised track, we compared CLM, MLM, and Explicit Sentence Compression (ESC) pre-training approaches joint trained with BT in the second stage of UNMT, respectively, and found that MLM and ESC had similar effects and were stronger than CLM. Moreover, the pre-training baseline of MLM was stronger than that of MASS. The combination of unsupervised training of DE-HSB and supervised training of EN-DE achieves the purpose of transfer learning, and the improvement is greater than 3 BLEU. Based on the conclusion of MLM and BT joint training on the UNMT Baseline, we also got a similar

[1]Our systems in unsupervised track are not a constrained unsupervised system due to the utilization of additional parallel corpora.

[2]The team name for PL→EN submission is "NICT-rui" in the OCELoT site to distinguish between different sub-teams.

Systems	DE→HSB			HSB→DE		
	Dev	Test	Official	Dev	Test	Official
UnsupSMT (Artetxe et al., 2018)	17.1	14.7	-	13.8	12.6	-
MASS baseline	29.8	26.0	-	31.4	27.3	-
UNMT baseline	31.1	27.2	-	31.3	27.2	-
+CLM finetune	29.2	25.6	-	28.6	24.5	-
+MLM finetune	32.4	28.3	-	32.4	27.3	-
+ESC finetune	32.1	28.3	-	32.2	27.8	-
EN-DE-HSB MUNMT baseline	29.3	25.6	-	30.0	26.2	-
++EN-DE NMT	33.6	29.3	-	33.6	29.6	-
++MLM finetune	35.1	30.5	28.6	34.9	30.7	28.6
++RAT + RABT + XBT	47.8	41.8	40.3	40.6	35.9	32.8

Table 3: DE↔HSB unsupervised performance (sacreBLEU score) for different models.

Systems	DE→HSB			HSB→DE		
	Dev	Test	Official	Dev	Test	Official
UNMT baseline	31.1	27.2	-	31.3	27.2	-
++MLM finetune	32.4	28.3	-	32.4	27.3	-
++DE-HSB NMT	59.9	53.0	52.5	61.6	53.1	54.6
++TLM finetune	60.2	53.2	-	61.4	52.7	-
++CFST	61.3	54.5	60.2	62.2	53.9	55.6
++D2GPo	61.4	54.6	60.4	62.9	54.5	56.6
Ensemble+Re-ranking	61.5	54.7	60.7	63.3	56.1	58.5
EN-DE-HSB MUNMT baseline	29.3	25.6	-	30.0	26.2	-
++EN-DE NMT + MLM finetune	35.1	30.5	28.6	34.9	30.7	28.6
++DE-HSB NMT	59.8	53.0	-	62.0	53.7	-

Table 4: DE↔HSB low-resource performance (sacreBLEU score) for different models.

trend on the MUNMT system. In the final system, the enhancement of RAT+RABT+XBT brought a BLEU increase of 11.7 and 4.2, respectively.

In the low-resource track, the model in the unsupervised track is used as the pre-trained model, and DE-HSB NMT and BT are jointly trained. Due to the DE-HSB parallel corpus, we can not only use MLM for monolingual pre-training, but also use TLM for cross-lingual pre-training. The addition of CFST and D2GPo further improves the effect of the model, indicating that these contributions are orthogonal. In addition, comparing UNMT with MUNMT given a parallel corpus, we found that although MUNMT used more data, it did not bring about a large enough effect improvement, so we will leave it for future research.

5 Conclusion

This paper describes SJTU-NICT's submission to the WMT20 news translation task. For three typical scenarios, we adopt different strategies. In this work, we not only study the pre-trained language model to enhance MT, but also consider the impact of document information on translation. We considered both the way of converting document alignment into sentence alignment and the use of BERT's NSP to recover the structure of documents. In addition, transfer learning from supervision is taken into account in unsupervised translation, and various means are used to enhance low-resource translation. Our systems performed strongly among all the submissions: we ranked 1st in PL→EN, EN→ZH, and DE→HSB respectively, and stayed Top-3 for the HSB→DE.

References

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018. Unsupervised statistical machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium. Association for Computational Linguistics.

Yehoshua Bar-Hillel. 1960. The present status of automatic translation of languages. In *Advances in computers*, volume 1, pages 91–163. Elsevier.

Rachel Bawden, Rico Sennrich, Alexandra Birch, and Barry Haddow. 2018. Evaluating discourse phenomena in neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1304–1313.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv:2004.05150*.

Stéphane Clinchant, Kweon Woo Jung, and Vassilina Nikoulina. 2019. On the use of bert for neural machine translation. *EMNLP-IJCNLP 2019*, page 108.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems*, pages 7059–7069.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G Carbonell, Quoc Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

William A Gale and Kenneth Church. 1993. A program for aligning sentences in bilingual corpora. *Computational linguistics*, 19(1):75–102.

Di He, Yingce Xia, Tao Qin, Liwei Wang, Nenghai Yu, Tie-Yan Liu, and Wei-Ying Ma. 2016. Dual learning for machine translation. In *Advances in neural information processing systems*, pages 820–828.

Junxian He, Jiatao Gu, Jiajun Shen, and Marc'Aurelio Ranzato. 2019. Revisiting self-training for neural sequence generation. *arXiv preprint arXiv:1909.13788*.

Shexia He, Zuchao Li, Hai Zhao, and Hongxiao Bai. 2018. Syntax for semantic role labeling, to be, or not to be. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2061–2071, Melbourne, Australia. Association for Computational Linguistics.

Matthew Honnibal and Ines Montani. 2017. spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. *To appear*, 7(1).

Kenji Imamura and Eiichiro Sumita. 2019. Recycling a pre-trained bert encoder for neural machine translation. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 23–31.

Sébastien Jean and Kyunghyun Cho. 2019. Context-aware learning for neural machine translation. *arXiv preprint arXiv:1903.04715*.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, et al. 2018. Marian: Fast neural machine translation in c++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121.

Dan Jurafsky. 2000. *Speech & language processing*. Pearson Education India.

Yunsu Kim, Yingbo Gao, and Hermann Ney. 2019. Effective cross-lingual transfer of neural machine translation models without shared vocabularies. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1246–1257.

Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. 2019. Reformer: The efficient transformer. In *International Conference on Learning Representations*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*, pages 177–180. Association for Computational Linguistics.

Olga Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky. 2019. Revealing the dark secrets of bert. *arXiv preprint arXiv:1908.08593*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. ALBERT: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*.

Zuchao Li, Jiaxun Cai, Shexia He, and Hai Zhao. 2018. Seq2seq dependency parsing. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3203–3214, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Zuchao Li, Rui Wang, Kehai Chen, Masao Utiyama, Eiichiro Sumita, Zhuosheng Zhang, and Hai Zhao. 2019a. Explicit sentence compression for neural machine translation. *arXiv preprint arXiv:1912.11980*.

Zuchao Li, Rui Wang, Kehai Chen, Masso Utiyama, Eiichiro Sumita, Zhuosheng Zhang, and Hai Zhao. 2020a. Data-dependent gaussian prior objective for language generation. In *International Conference on Learning Representations*.

Zuchao Li, Hai Zhao, Rui Wang, Masao Utiyama, and Eiichiro Sumita. 2020b. Reference language based unsupervised neural machine translation. *arXiv preprint arXiv:2004.02127*.

Zuchao Li, Junru Zhou, Hai Zhao, and Rui Wang. 2019b. Cross-domain transfer learning for dependency parsing. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 835–844. Springer.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Marco Lui and Timothy Baldwin. 2012. langid. py: An off-the-shelf language identification tool. In *Proceedings of the ACL 2012 system demonstrations*, pages 25–30.

Ruixuan Luo, Jingjing Xu, Yi Zhang, Xuancheng Ren, and Xu Sun. 2019. Pkuseg: A toolkit for multi-domain chinese word segmentation. *CoRR*, abs/1906.11455.

Ying Luo and Hai Zhao. 2020. Bipartite flat-graph network for nested named entity recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6408–6418, Online. Association for Computational Linguistics.

Sameen Maruf, Fahimeh Saleh, and Gholamreza Haffari. 2019. A survey on document-level machine translation: Methods and evaluation. *arXiv preprint arXiv:1912.08494*.

Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. 2016. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Sinno Jialin Pan and Qiang Yang. 2009. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359.

Michael Paul, Hirofumi Yamamoto, Eiichiro Sumita, and Satoshi Nakamura. 2009. On the importance of pivot language selection for statistical machine translation. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*, pages 221–224, Boulder, Colorado. Association for Computational Linguistics.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting bleu scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191.

Stephen Robertson and Hugo Zaragoza. 2009. *The probabilistic relevance framework: BM25 and beyond*. Now Publishers Inc.

Yves Scherrer, Jörg Tiedemann, and Sharid Loáiciga. 2019. Analysing concatenation approaches to document-level nmt in two different domains. In *Proceedings of the Fourth Workshop on Discourse in Machine Translation (DiscoMT 2019)*, pages 51–61.

H Scudder. 1965. Probability of error of some adaptive pattern-recognition machines. *IEEE Transactions on Information Theory*, 11(3):363–371.

Rico Sennrich. 2018. Why the time is ripe for discourse in machine translation. In *Second Workshop on Neural Machine Translation and Generation*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725.

Haipeng Sun, Rui Wang, Kehai Chen, Masao Utiyama, Eiichiro Sumita, and Tiejun Zhao. 2020a. Self-training for unsupervised neural machine translation in unbalanced training data scenarios. *arXiv preprint arXiv:2004.04507*.

Haipeng Sun, Rui Wang, Kehai Chen, Masao Utiyama, Eiichiro Sumita, and Tiejun Zhao. 2020b. Unsupervised neural machine translation with cross-lingual language representation agreement. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:1170–1182.

Lisa Torrey and Jude Shavlik. 2010. Transfer learning. In *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*, pages 242–264. IGI global.

Masao Utiyama and Hitoshi Isahara. 2007. A comparison of pivot methods for phrase-based statistical machine translation. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 484–491, Rochester, New York. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Elena Voita, Pavel Serdyukov, Rico Sennrich, and Ivan Titov. 2018. Context-aware neural machine translation learns anaphora resolution. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1264–1274.

Xinyi Wang, Jason Weston, Michael Auli, and Yacine Jernite. 2019. Improving conditioning in context-aware sequence to sequence models. *arXiv preprint arXiv:1911.09728*.

Hua Wu and Haifeng Wang. 2007. Pivot language approach for phrase-based statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 856–863, Prague, Czech Republic. Association for Computational Linguistics.

Fengshun Xiao, Jiangtong Li, Hai Zhao, Rui Wang, and Kehai Chen. 2019. Lattice-based transformer encoder for neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3090–3097, Florence, Italy. Association for Computational Linguistics.

Deyi Xiong, Yang Ding, Min Zhang, and Chew Lim Tan. 2013. Lexical chain based cohesion models for document-level statistical machine translation. In *Proceedings of the 2013 conference on empirical methods in Natural Language Processing*, pages 1563–1573.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. XLNet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5753–5763.

David Yarowsky. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *33rd annual meeting of the association for computational linguistics*, pages 189–196.

Huan Zhang and Hai Zhao. 2018. Minimum divergence vs. maximum margin: an empirical comparison on seq2seq models. In *International Conference on Learning Representations*.

Zhuosheng Zhang, Jiangtong Li, Pengfei Zhu, Hai Zhao, and Gongshen Liu. 2018. Modeling multi-turn conversation with deep utterance aggregation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3740–3752, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Junru Zhou and Hai Zhao. 2019. Head-Driven Phrase Structure Grammar parsing on Penn Treebank. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2396–2408, Florence, Italy. Association for Computational Linguistics.

Jinhua Zhu, Yingce Xia, Lijun Wu, Di He, Tao Qin, Wengang Zhou, Houqiang Li, and Tieyan Liu. 2020. Incorporating bert into neural machine translation. In *International Conference on Learning Representations*.

Xiaojin Zhu and Andrew B Goldberg. 2009. Introduction to semi-supervised learning. *Synthesis lectures on artificial intelligence and machine learning*, 3(1):1–130.

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer learning for low-resource neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1568–1575.

Combination of Neural Machine Translation Systems at WMT20

Benjamin Marie **Raphael Rubino** **Atsushi Fujita**
National Institute of Information and Communications Technology
3-5 Hikaridai, Seika-cho, Soraku-gun, Kyoto, 619-0289, Japan
{bmarie, raphael.rubino, atsushi.fujita}@nict.go.jp

Abstract

This paper presents neural machine translation systems and their combination built for the WMT20 English↔Polish and Japanese→English translation tasks. We show that using a Transformer Big architecture, additional training data synthesized from monolingual data, and combining many NMT systems through n-best list reranking improve translation quality. However, while we observed such improvements on the validation data, we did not observe similar improvements on the test data. Our analysis reveals that the presence of translationese texts in the validation data led us to take decisions in building NMT systems that were not optimal to obtain the best results on the test data.

1 Introduction

This paper describes the neural machine translation (NMT) systems and their combination built for a participation of the National Institute of Information and Communications Technology (NICT) in the WMT20 shared News Translation Task.[1] We participated in three translation directions: Japanese→English (Ja→En), English→Polish (En→Pl), and Polish→English (Pl→En). All our systems are *constrained*, i.e., we used only the parallel and monolingual data provided by the organizers to train and tune them, and validated/selected our best systems exclusively using the official validation data provided by the organizers. We trained NMT systems with several different frameworks and architectures, and combined them, for each translation direction, through n-best list reranking using informative features as proposed by Marie and Fujita (2018). This simple combination method, associated with the exploitation of large tagged back-translated monolingual data, improved BLEU scores on the official validation data provided by the organizers. However, we did not observe these improvements on the test data for which our baseline systems remained the best. While we have rigorously selected our systems according to their performance on the validation data, the analysis of our results reveal how easily we would have been able to achieve BLEU scores among the best submissions by choosing/selecting our best systems according to their performance on the test data, as encouraged by the WMT submission process (Section 2).

The remainder of this paper is organized as follows. In Section 2, we briefly describe the WMT20 translation task. In Section 3, we introduce the data pre-processing and cleaning. In Section 4, we describe the details of our NMT systems' architectures and frameworks. In Section 5, we describe two different strategies that we used to augment the training parallel data of our systems: parallel data extraction from monolingual data and backward/forward translations. Then, the combination of our NMT systems is described in Section 6. Empirical results produced with our systems on the validation and test data are presented in Section 7. We propose an analysis in Section 8 to better understand why our best systems on the validation data are significantly worse on the test data. Section 9 concludes this paper.

2 Description of the Task

The task is to translate texts in the news domain. For this purpose, news articles were sampled from online newspapers from September–November 2019. The sources of the test data are original texts whereas the targets are human-produced translations, i.e., participants are not asked to translate translationese texts unlike past WMT translation tasks. Although organizers also mentioned that the provided validation data were

[1] The team ID of our participation is "NICT_Kyoto".

Proceedings of the 5th Conference on Machine Translation (WMT), pages 230–238
Online, November 19–20, 2020.

created in the same way as the test data, they were actually made half of translationese texts and half of original texts. For the Inuktitut→English translation task, source texts to translate in the test data were only translationese texts.

Training parallel and monolingual data were provided for all language pairs. Participants were asked to mention whether they used additional external data. We chose to participate in the *constrained* settings using only the provided data to train our MT systems. Validation data were also provided for each language pair. We used the entire data to keep it sufficiently large for validation purposes, even though half of it was made up of translationese texts.

For collecting submissions, organizers relied on a new framework: `Ocelot`.[2] Each participant was allowed to submit up to 8 submissions per account but was not limited in the number of accounts. Upon submission, `Ocelot` shows the corresponding chrF and BLEU scores computed using reference translations that were not released during the competition. Participants could then rely on these scores to select and validate their best system on the test data.[3] We chose to ignore these scores obtained on the test data to remain in a much more realistic scenario where we do not have access to reference translations, i.e., we relied only on the validation data to select our primary submission.

Primary submissions selected by the participants were then evaluated by humans which is the official evaluation for WMT translation tasks.

3 Data Pre-processing and Cleaning

3.1 Data

As parallel data to train our systems, we used all the provided data for all our targeted translation directions, except the "Wiki Titles"[4] corpus. As English monolingual data, we used all the provided data, but sampled only 200M lines from the "Common Crawl" corpora, except the "News Discussions" and "Wiki Dumps" corpora. For all other languages, we used all the provided monolingual corpora but also sampled only 200M lines from the "Common Crawl" corpora.

To tune/validate and evaluate our systems, we used the official validation and test data provided by the organizers.

Language pair	#sent. pairs	#tokens	
En–Pl	8.7M	239.5M (En)	310.0M (Pl)
En–Ja	15.2M	394.5M (En)	380.6M (Ja)

Table 1: Statistics of our pre-processed parallel data.

3.2 Pre-processing and Cleaning

Since some corpora were crawled from the Web and therefore potentially very noisy, we first performed language identification on all the data to keep only lines that have a high probability of being in the right language. We used `fastText` (Bojanowski et al., 2017) and its large model for language identification.[5] We only retained sentences that have a probability higher than 0.75 to be in the right language. For the parallel data, if at least one side of each sentence pair did not match this criteria, we removed the pair from the corpus.

We used Moses (Koehn et al., 2007) punctuation normalizer, tokenizer, and truecaser for English and Polish. The truecaser was trained on the News Crawl 2019 corpora. Truecasing was then performed on all the tokenized data. Then, for the Pl–En language pair, we jointly learned 32k BPE operations (Sennrich et al., 2016b) on the concatenation of English and Polish News Crawl 2019 corpora. We performed sub-word segmentation using this vocabulary on the Polish and English parallel and monolingual data. For the Ja–En language pair, we independently learned 32k BPE operations on the English News Crawl 2019 corpus for English, 32k sentence piece (Kudo and Richardson, 2018) operations on the Japanese News Crawl 2019 corpus for Japanese, and then applied the operations to perform sub-word segmentation on the data in their respective language.

For further cleaning of the data, we applied the script "clean-corpus-n.perl" from Moses to remove empty lines and sentences longer than 120 sub-word tokens. Tables 1 and 2 present the statistics of the parallel and monolingual data, respectively, after pre-processing.

[2] https://ocelot.mteval.org/

[3] We can read in the "competition updates" that this behavior was encouraged by the organizers: "Also added chrF computation to give you more data points for your primary submission selection. Submissions remain ordered by decreasing SacreBLEU score."

[4] It contains only very short segments that are not sentences. We therefore assume to be of limited use in NMT.

[5] https://fasttext.cc/docs/en/language-identification.html

Language	#lines	#tokens
En (En–Pl)	328M	7.9B
En (En–Ja)	328M	7.7B
Ja	184M	4.8B
Pl	137M	3.2B

Table 2: Statistics of our pre-processed monolingual data.

4 NMT systems

4.1 Architectures

Transformer Base and Big For our NMT systems, we chose the Transformer architecture (Vaswani et al., 2017). In this paper, we refer to Transformer Base and Big as the "*base*" and "*big*" configurations from Vaswani et al. (2017)'s paper. The architecture differences are as follows:

- Base: 512 embedding dimensions, 2,048 dimensions for the feed-forward, and 8 heads
- Big: 1024 embedding dimensions, 4,096 dimensions for the feed-forward, and 16 heads

Highway Transformer Residual connections (RCs) (Srivastava et al., 2015a; He et al., 2016) have been shown to increase forward and backward information flow in deep neural networks (Hardt and Ma, 2017) and thus are a crucial component of the Transformer architecture. Removing them has a negative impact on training and on the overall performances of the resulting model (Bapna et al., 2018). However, incorporating RCs through the addition operation as it is commonly done in the Transformer network does not allow for a distribution of weights between carrying or transforming the input. An alternative, inspired by the *Highway Network* (Srivastava et al., 2015b) and implemented within the Transformer by Chai et al. (2020), includes a trainable gating mechanism that regulates the information flow. We applied a few modifications to the implementation proposed in Chai et al. (2020): removing all layer normalization operations, adding depth-aware parameter initialization (Junczys-Dowmunt, 2019; Zhang et al., 2019), and initializing biases so that the residual blocks are initially forced to carry information rather than transforming it (Srivastava et al., 2015b).

4.2 Frameworks and Settings

`Marian` Our Models trained with the `Marian` toolkit (Junczys-Dowmunt et al., 2018) were only based on Transformer *Base*. We set the dropout at 0.1 and used the mini-batch-fit option of `Marian` to have batches as large as allowed by the size of the GPU memory. We used ReLU activation functions and optimized the models using the Adam optimizer with parameters $\beta_1 = 0.9$, $\beta_2 = 0.98$, and $\epsilon = 1e^{-9}$, a learning rate initialized at $3e^{-4}$, following a linear warm-up during 16k updates and decaying based on the inverse square root of the update number. Label smoothing was set to 0.1. During training, mean cross-entropy was evaluated on the entire validation data every 5,000 mini-batch updates and training was stopped after 5 consecutive times without an improvement of the mean cross-entropy. Then, we selected the best model that yielded the best BLEU score on the validation data. For decoding, we fixed the beam size at 12 and the length normalization at 1.0.

`Fairseq` Models trained with the `Fairseq` toolkit were based on Transformer *base* and Transformer *big*. The former used a dropout rate of 0.1 and batches containing approximately 12k tokens with parameters updated every 2 batches. The latter used a dropout rate of 0.3 and batches containing approximately 8k tokens with parameters updated every 8 batches. Both configurations shared decoder input and output embeddings, trained with half-precision float numbers, used ReLU activation functions, were optimized using the Adam optimizer with parameters $\beta_1 = 0.9$, $\beta_2 = 0.98$, and $\epsilon = 1e^{-9}$, a learning rate initialized at $1.7e^{-7}$, following a linear warm-up during 4k updates until reaching $5e^{-4}$ and decaying based on the inverse square root of the update number. Label smoothing with a parameter 0.1 was applied during training. Whereas *base* models were trained for 200 epochs, *big* models were trained for 100 epochs. The entire validation data was used for evaluation every epoch, while the best BLEU scores on this data allow for checkpoint saving. The parameters for decoding were fixed: a beam size of 4 and a length penalty of 0.6.

5 Training Data Augmentation

5.1 Parallel Data Alignment

We extracted additional training parallel data from the News Crawl monolingual corpora with the following procedure:

1. Jointly train bilingual word embeddings on

Configuration	En–Pl			Ja→En	#sent. pairs
	En→Pl	Pl→En	#sent. pairs		
w/o NC	26.3	30.4	0	20.3	0
w/ NC	26.4	30.6	257.4k	20.3	244.1k

Table 3: Results obtained on the validation data with `Fairseq` Big with and without using the additional parallel data extracted from the News Crawl monolingual corpora, denoted "NC." The columns "#sent. pairs" indicate how many sentence pairs were extracted from the News Crawl monolingual corpora.

the provided parallel data using `Bivec` (Luong et al., 2015).

2. Make all possible bilingual sentence pairs from News Crawl corpora in the source and target languages.

3. For each sentence pair, compute the similarity between the source and target sentences using the bilingual word embeddings trained with `Bivec` simply by measuring cosine similarity over the averaged word embeddings in each sentence as proposed by Artetxe and Schwenk (2019).[6]

4. Finally, keep only the sentence pairs with a score higher than a threshold among $\{1.0, 1.0025, 1.05\}$ and select the value that results in the sentence pairs leading to the highest BLEU score on the validation data when mixing the selected sentence pairs with the original parallel data for training NMT.

Table 3 gives an overview of the results obtained with the additional sentence pairs extracted from News Crawl. We did not observe significant improvements as we could only extract a very small amount of useful sentence pairs. Nevertheless, we decided to keep these additional data to train our other NMT systems, since it did not appear harmful according to BLEU. However, as we report in Section 8, it was not the optimal choice to obtain the best results on the test data.

5.2 Backward and Forward Translation of Monolingual Data

Parallel data for training NMT can be augmented with synthetic parallel data, generated through a so-called back(ward)-translation, to significantly improve translation quality (Bertoldi and Federico, 2009; Bojar and Tamchyna, 2011; Sennrich et al., 2016a). We used the `Fairseq` Big system, trained on the provided parallel data and the aligned News Crawl sentence pairs, to translate target monolingual sentences into the source language. Then, these back-translated sentences were simply mixed with the original parallel data, putting the synthetic side on the source side, to train from scratch a new NMT system.

We also experimented with forward translation, i.e., with the synthetic part on the target side, and tagged back-translation (Caswell et al., 2019), which simply adds a tag at the beginning of each back-translation, as it has shown to lead to better results, especially when translating texts in their original language (Marie et al., 2020).

For English, we translated 50M sentences made of the entire News Crawl 2019 corpus and randomly added sentences from News Crawl 2018 corpus until we have 50M sentences. For Polish and Japanese, we translated the entire News Crawl corpora and added sentences from the Common Crawl corpus until we have 50M sentences.

For each configuration, i.e., back-translation, tagged back-translation, and forward translation, we also experimented with sub-samples of 12.5M (only with `Marian`), and 25M synthetic sentence pairs, in addition to using the entire 50M sentence pairs, for retraining the NMT systems. Table 4 gives an overview of the results for each configuration obtained on the validation data.

All configurations using back-translations (BT) and tagged back-translations (TBT) were better than the baseline system as expected. We also observed very small differences in BLEU when increasing the size of the back-translated data.

TBT improves over BT as expected (Caswell et al., 2019), but only for Pl→En and Ja→En. On the other hand, using forward translations significantly decreased BLEU scores, as expected, since it introduces NMT translations to the target side of the training data (Bogoychev and Sennrich, 2019), but again only for Pl→En and Ja→En. Our results for En→Pl across all configurations remained similar, which defies the findings of previous work on back-translation and forward translation. We give

[6] We used the "Ratio" version of the scoring function.

System	#sent. pairs	En→Pl	Pl→En	Ja→En
Marian Base	0	24.4	29.1	18.1
BT	12.5M	26.1	32.1	21.1
	25M	26.2	32.1	21.2
	50M	26.3	31.7	21.1
TBT	12.5M	26.1	30.3	21.1
	25M	26.1	32.3	21.2
	50M	26.3	32.2	21.4
FWD	12.5M	26.3	29.7	18.3
	25M	26.4	29.5	17.4
	50M	26.3	29.6	16.4

Table 4: Results of Marian Base on the validation data obtained using synthetic parallel data as back-translations (BT), tagged back-translations (TBT) or forward translations (FWD).

Feature	Description
NMT models	Scores given by each NMT model
LEX	Sentence-level translation probabilities, for both translation directions
LM	Scores given by a 4-gram language model trained on all the monolingual corpora in the target language
LEN	Difference between the length of the source sentence and the length of the translation hypothesis, and its absolute value

Table 5: Set of features used by our reranking systems. The "Feature" column refers to the same feature used in Marie and Fujita (2018). The numbers between parentheses indicate the number of scores in each feature set.

some plausible explanations for this peculiarity in our analysis in Section 8.

6 Combination of NMT systems

Our primary submissions for the tasks were the result of a simple combination of all our NMT systems through n-best list reranking. As demonstrated by Marie and Fujita (2018), it can significantly improve translation quality, even when there is a large difference in translation quality between the combined systems. Following Marie and Fujita (2018), our system combination works as follows.

6.1 Generation of n-best Lists

We first independently generated the 100-best translation hypotheses from each of all our NMT models, and additional 12-best, with Marian, or 4-best with Fairseq. We then merged all these lists generated by different systems, without removing duplicated hypotheses.

6.2 Reranking Framework and Features

We rescored all the hypotheses in the list with a reranking framework using features to better model the fluency and the adequacy of each hypothesis. This method can find a better hypothesis in these merged n-best lists than the one-best hypothesis originated by the individual systems. We chose KB-MIRA (Cherry and Foster, 2012) as a rescoring framework and used a subset of the features proposed in Marie and Fujita (2018). All the following features we used are described in detail by Marie and Fujita (2018). As listed in Table 5, it includes all scores given by all our NMT models. We computed sentence-level translation probabilities using the lexical translation probabilities learned by mgiza on all the parallel training data of our NMT systems. One 4-gram language model trained on the target language model was also used. To account for hypotheses length, we added the difference, and its absolute value, between the number of tokens in the translation hypothesis and the source sentence. The reranking framework was trained on n-best lists generated by decoding the entire validation data.

7 Results

Our main results are presented in the Table 6. According to the validation data, Fairseq Base is as good as, or better than, Marian Base. Given this observation, we trained Fairseq Big and obtained even better results on the validation data. BLEU scores are improved by up to 4.1 BLEU points when using tagged back-translations (TBT) on the validation data. Overall, the best system is, as expected, the Reranker combining all our systems with additional features. How-

System	En→Pl		Pl→En		Ja→En	
	Validation	Test	Validation	Test	Validation	Test
Marian Base	24.4	21.2	29.1	31.9	18.1	19.1
Fairseq Base	24.4	21.8	30.3	32.4	19.4	18.7
Fairseq Big	26.4	21.9	30.9	31.5	20.4	19.3
Fairseq Big TBT	28.5	23.1	35.0	31.8	23.3	19.9
Reranker∗	29.9	24.9	36.5	32.3	25.5	22.8

Table 6: Results of our main systems. Marian Base and Fairseq Base use the same training data and architecture. Fairseq Big uses Transformer Big and the additional training data extracted from News Crawl corpora. Fairseq Big TBT is retrained from scratch on the tagged back-translations generated by Fairseq Big. The system denoted with an "∗" is our primary system.

ever, surprisingly, the results on the test data exhibited a significantly different pattern. For instance, Marian Base performed very closely to Fairseq Big on the test data. Even more strikingly, we observed only small differences in BLEU between Fairseq Big and Fairseq TBT. For instance, for Pl→En, while we have 4.1 BLEU points of improvements on the validation data, we have only 0.3 BLEU points of improvements on the test data. Also for this translation direction, Reranker outperforms Marian Base by 7.4 BLEU points on the validation data but by only 0.4 BLEU points on the test data.

To better understand the lack of correlation between our results on validation and test data, we propose an analysis in the next section.

8 Analysis

Table 7 presents all our results for En↔Pl on the validation data, separating the part in the original and non-original languages, and the test data.

One obvious observation from these results is that the BLEU scores on the test data are all very close to the score of the validation data in original language (Orig.). On the other hand, we also observe that the ranking of the systems given the BLEU score on the entire validation data does not correlate well with the ranking of the systems given the BLEU score on the validation data in the original language. It means that the translationese texts in the validation data had a negative impact on all our decisions for selecting the best framework, architecture, additional parallel sentences, and so forth, and that we could potentially had better results by taking our decisions by using only the original texts in the validation data.

Translationese texts are particularly harmful for training a Reranker, as we can observe for Pl→En. Using them as training data for the Reranker leads to significantly lower BLEU scores (#14) while training it only on the original texts of the validation data leads to our best BLEU score (#15). For this translation direction, we also observe large improvements of BLEU thanks to back-translations on the validation data that comes mainly from the translationese texts while translation quality drops when translating the original texts in the validation and test data, as expected. This was compensated by using tagged back-translations (#10-12) as suggested by Marie et al. (2020).

Our observations are very different for the reverse translation direction. For En→Pl, training Reranker on the entire validation data leads to the best BLEU score, and it drops only slightly when training only on the translationese texts. Even more surprisingly, using back-translations improves BLEU scores for both original and translationese texts while using tagged back-translations (#12) leads to BLEU scores identical to those obtained by using back-translations (#9) for the original texts. These peculiarities observed for Pl→En, associated with our observations in Section 5.2 that forward translations improves BLEU, are in contradiction with the findings in previous work as follows.

- Back-translations should decrease BLEU scores for original texts (Edunov et al., 2020).
- Tagged back-translations should improve BLEU scores for original texts (Marie et al., 2020).
- Forward translation should lead to lower BLEU scores for translationese texts (Bogoychev and Sennrich, 2019).

A possible explanation is that the texts denoted as "original" in the validation data and the test data,

#	System	Arch.	NC	BT	TBT	En→Pl Validation Orig.	En→Pl Validation Non-orig.	En→Pl Validation All	En→Pl Test	Pl→En Validation Orig.	Pl→En Validation Non-orig.	Pl→En Validation All	Pl→En Test
						Individual Systems							
1	Marian	Base				21.4	28.7	24.4	21.2	32.0	26.7	29.1	32.3
2	Marian	Base	✓			21.5	28.5	24.8	21.7	32.3	27.5	29.7	31.9
3	Fairseq	Base				20.9	28.7	24.4	21.8	33.1	27.9	30.3	32.4
4	Fairseq	Highway				21.1	29.6	25.0	21.8	32.6	28.5	30.6	32.6
5	Fairseq	Big				22.7	30.7	26.3	22.3	31.2	30.3	30.7	32.5
6	Fairseq	Big	✓			22.6	31.2	26.4	21.9	31.9	29.5	30.9	31.5
7	Marian	Big	✓	✓		22.1	31.3	26.3	21.9	29.1	34.1	32.0	29.5
8	Fairseq	Base	✓	✓		22.2	32.1	26.8	22.2	29.7	34.9	32.9	29.7
9	Fairseq	Big	✓	✓		23.6	34.4	28.5	23.7	30.2	36.5	33.9	29.5
10	Marian	Big	✓		✓	22.1	31.1	26.2	22.1	32.1	32.9	32.5	31.8
11	Fairseq	Base	✓		✓	22.3	32.0	26.7	22.2	31.8	34.0	33.2	31.7
12	Fairseq	Big	✓		✓	23.6	34.6	28.5	23.1	32.4	37.1	35.0	31.8
						System Combination							
13	Reranker∗					25.6	35.2	29.9	24.9	33.1	38.7	36.5	32.3
14	Reranker Non-orig.					23.2	35.3	29.3	23.1	28.8	39.6	34.7	28.4
15	Reranker Orig.					25.4	33.9	29.1	24.7	35.0	33.8	34.4	34.8

Table 7: The system denoted with an "∗" is our primary system. The column "Arch." stands for the Transformer architecture, "NC" indicates the use of the News Commentary Corpus, "BT" and "TBT" indicate the use of back-translation and tagged back-translation, respectively. "Reranker Non-orig." and "Reranker Orig." are variants of Reranker that are trained on the validation data using only the part in the non-original and original languages, respectively, while Reranker, our primary system, was trained on the entire validation data.

that were prepared similarly, do have some characteristics of translationese that may come from the translation of texts not in their original language or the use of MT followed by post-editing. Comparing our results with the results of other participants will help us test this assumption.

9 Conclusion

We participated in three translation directions and for all of them we did experiments with several frameworks and architectures, also exploiting additional synthetic parallel data made from monolingual data. Combining all our systems led to significantly better BLEU scores on the validation data. However, our analysis revealed that the presence of translationese texts in the validation data led us to take sub-optimal choices that prevented us from obtaining significantly better BLEU scores on the test data. Selecting/validating systems on the test data should not be possible, or at least not an option. We thus suggest organizers to provide validation data that better matches the characteristics of the test data, e.g., removing translationese texts if none are in the test data.

Acknowledgments

We would like to thank Reviewer 1 for the detailed corrections and suggestions.

References

Mikel Artetxe and Holger Schwenk. 2019. Margin-based parallel corpus mining with multilingual sentence embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3197–3203, Florence, Italy. Association for Computational Linguistics.

Ankur Bapna, Mia Chen, Orhan Firat, Yuan Cao, and Yonghui Wu. 2018. Training deeper neural machine translation models with transparent attention. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3028–3033, Brussels, Belgium. Association for Computational Linguistics.

Nicola Bertoldi and Marcello Federico. 2009. Domain adaptation for statistical machine translation with monolingual resources. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 182–189, Athens, Greece. Association for Computational Linguistics.

Nikolay Bogoychev and Rico Sennrich. 2019. Domain, translationese and noise in synthetic data for neural machine translation. *arXiv preprint arXiv:1911.03362*.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Ondřej Bojar and Aleš Tamchyna. 2011. Improving translation model by monolingual data. In *Proceedings of the Sixth Workshop on Statistical Machine*

Translation, pages 330–336, Edinburgh, Scotland. Association for Computational Linguistics.

Isaac Caswell, Ciprian Chelba, and David Grangier. 2019. Tagged back-translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 53–63, Florence, Italy. Association for Computational Linguistics.

Yekun Chai, Shuo Jin, and Xinwen Hou. 2020. Highway transformer: Self-gating enhanced self-attentive networks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6887–6900, Online. Association for Computational Linguistics.

Colin Cherry and George Foster. 2012. Batch tuning strategies for statistical machine translation. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 427–436, Montréal, Canada. Association for Computational Linguistics.

Sergey Edunov, Myle Ott, Marc'Aurelio Ranzato, and Michael Auli. 2020. On the evaluation of machine translation systems trained with back-translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2836–2846, Online. Association for Computational Linguistics.

Moritz Hardt and Tengyu Ma. 2017. Identity matters in deep learning. In *Proceedings of the 5th International Conference on Learning Representations (ICLR)*, Toulon, France.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the 29th IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, Las Vegas, USA.

Marcin Junczys-Dowmunt. 2019. Microsoft translator at WMT 2019: Towards large-scale document-level neural machine translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 225–233, Florence, Italy. Association for Computational Linguistics.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Bilingual word representations with monolingual quality in mind. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 151–159, Denver, USA. Association for Computational Linguistics.

Benjamin Marie and Atsushi Fujita. 2018. A smorgasbord of features to combine phrase-based and neural machine translation. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Papers)*, pages 111–124, Boston, USA. Association for Machine Translation in the Americas.

Benjamin Marie, Raphael Rubino, and Atsushi Fujita. 2020. Tagged back-translation revisited: Why does it really work? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5990–5997, Online. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Rupesh K Srivastava, Klaus Greff, and Jürgen Schmidhuber. 2015a. Training very deep networks. In *Advances in Neural Information Processing Systems 28*, pages 2377–2385, Montréal, Canada. Curran Associates, Inc.

Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. 2015b. Highway Networks. *arXiv preprint arXiv:1505.00387*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, Long Beach, USA. Curran Associates, Inc.

Biao Zhang, Ivan Titov, and Rico Sennrich. 2019. Improving deep transformer with depth-scaled initialization and merged attention. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 898–909, Hong Kong, China. Association for Computational Linguistics.

WeChat Neural Machine Translation Systems for WMT20

Fandong Meng, Jianhao Yan, Yijin Liu, Yuan Gao, Xianfeng Zeng,
Qinsong Zeng, Peng Li, Ming Chen, Jie Zhou, Sifan Liu and Hao Zhou
WeChat AI, Tencent, China
{fandongmeng,elliottyan,yijinliu,masongao,xianfzeng}@tencent.com
{qinzzeng,patrickpli,mingchen,withtomzhou,stephenliu,harveyzhou}@tencent.com

Abstract

We participate in the WMT 2020 shared news translation task on Chinese→English. Our system is based on the Transformer (Vaswani et al., 2017) with effective variants and the DTMT (Meng and Zhang, 2019) architecture. In our experiments, we employ data selection, several synthetic data generation approaches (i.e., back-translation, knowledge distillation, and iterative in-domain knowledge transfer), advanced finetuning approaches and self-bleu based model ensemble. Our constrained Chinese→English system achieves 36.9 case-sensitive BLEU score, which is the highest among all submissions.

1 Introduction

Our WeChat AI team participates in the WMT 2020 shared news translation task on Chinese→English. In this year's translation task, we mainly focus on exploiting several effective model architectures, better data augmentation, training and model ensemble strategies.

For model architectures, we mainly exploit two different architectures in our approaches, namely Transformers and RNMT. For Transformers, we implement the Deeper transformer with Pre-Norm, the Wider Transformer with a larger filter-size and the average attention based transformer (Zhang et al., 2018). For the RNMT, we use the deep transition based DTMT (Meng and Zhang, 2019) model. We finally ensemble four kinds of models in our system.

For synthetic data generation, we explore various methods for out-of-domain and in-domain data generation. For out-of-domain data generation, we explore the back-translation method (Sennrich et al., 2016a) to leverage the target side monolingual data and the knowledge distillation method (Kim and Rush, 2016) to leverage the source side of golden parallel data. For in-domain data generation, we employ iterative in-domain knowledge transfer to leverage the source-side monolingual data and golden parallel data. Furthermore, data augmentation methods, including noisy fake data (Wu et al., 2019) and sampling (Edunov et al., 2018), are used for training more robust NMT models.

For training strategies, we mainly focus on the parallel scheduled sampling (Mihaylova and Martins, 2019; Duckworth et al., 2019), the target denoising and minimum risk training (Shen et al., 2016; Wang and Sennrich, 2020) algorithm for in-domain finetuning.

We also exploit a self-bleu (Zhu et al., 2018) based model ensemble approach to enhance our system. As a result, our constrained Chinese→English system achieves the highest case-sensitive BLEU score among all submitted systems.

In the remainder of this paper, we start with an overview of model architectures in Section 2. Section 3 describes the details of our systems and training strategies. Then Section 4 shows the experimental settings and results. Finally, we conclude our work in Section 5.

2 Model Architectures

In this section, we first describe the model architectures we use in the Chinese→English Shared Task, including the Transformer-based (Vaswani et al., 2017) models and RNN-based (Bahdanau et al., 2014; Meng and Zhang, 2019) models.

2.1 Deeper Transformer

As shown in previous studies (Wang et al., 2019; Sun et al., 2019), deeper Transformers with pre-norm outperform its shallow counterparts on various machine translation benchmarks. In their work, increasing the encoder depth significantly improves the model performance, while they only introduce mild overhead in terms of speed in training and

Proceedings of the 5th Conference on Machine Translation (WMT), pages 239–247
Online, November 19–20, 2020.

inference, compared with increasing the decoder side depth.

Hence, we train deeper Transformers with a deep encoder aiming for a better encoding representation. In our experiments, we mainly adopt two settings, with the hidden size 512 (Base) and 1024 (Large). We adopt a 30-layer encoder for Base models and 20/24-layer encoders for Large models. Further increasing the encoder depth does not lead to a significant BLEU improvement. To keep the total trainable parameters the same among models, the filter sizes of Base and Large models are 16384 and 4096, respectively. For training, the batch size is 4,096 tokens per GPU, and we train each model using 8 NVIDIA V100 GPUs for about 7 days.

2.2 Wider Transformer

Inspired by last year's Baidu system (Sun et al., 2019), we also train Wider Transformers with a larger inner dimension of the Feed-Forward Network than the standard Transformer Large system. Specifically, two settings are used in our experiments. With a filter size of 15,000, we set the number of encoder layers to 10, and with a filter size of 12,288, we set the number of encoder layers to 12. The number of total trainable parameters of the Wider Transformer is kept approximately the same as our Deeper Transformers.

In our experiments, we also set the batch size to be 4,096 and train the Wider Transformers with 8 NVIDIA V100 GPUs for about 7 days.

2.3 Average Attention Transformer

To introduce more diversity in our Transformer models, we use Average Attention Transformer (AAN) (Zhang et al., 2018) as one of our candidate architectures. The Average Attention Transformer replaces the decoder self-attention module in auto-regressive order with a simple average attention, and introduces almost no loss in model performance.

We believe that even though the performance of AAN does not drop in terms of BLEU, the output distributions of AAN networks should be different from the output distributions of original Transformers, which brings diversity for the final ensemble. This also complies with our findings in self-bleu experiments (Section 3.6).

In practice, AAN models are trained for both the Wider Transformer and Deeper Transformer. The batch size and other hyper-parameters are kept the same as its non-AAN counterpart.

2.4 DTMT

DTMT (Meng and Zhang, 2019) is the recently proposed deep transition RNN-based model for Neural Machine Translation, whose encoder and decoder are composed of the well-designed transition blocks, each of which consists of a linear transformation enhanced GRU (L-GRU) followed by several transition GRUs (T-GRUs). DTMT enhances the hidden-to-hidden transition with multiple non-linear transformations, as well as maintains a linear transformation path throughout this deep transition by the well-designed linear transformation mechanism to alleviate the vanishing gradient problem. This architecture has demonstrated its superiority over the conventional Transformer model and stacked RNN-based models in NMT (Meng and Zhang, 2019), and also achieves surprising performances on other NLP tasks, such as sequence labeling (Liu et al., 2019) and aspect-based sentiment analysis (Liang et al., 2019).

In our experiments, we use the bidirectional deep transition encoder, where each directional deep transition block consists of 1 L-GRU and 4 T-GRU. The decoder contains a query transition block and the decoder transition block, each of which consists of 1 L-GRU and 4 T-GRU. Therefore the DTMT consists of a 5 layer encoder and a 10 layer decoder, with a hidden size of 1,024. We use 8 NVIDIA V100 GPUs to train each model for about three weeks and the batch size is set to 4,096 tokens per GPU.

3 System Overview

In this section, we describe our system used in the WMT 2020 news shared task.

Figure 1 depicts the overview of our Wechat NMT. Our system can be divided into four parts, namely data filtering, synthetic data generation, in-domain finetuning, and ensemble. The synthetic generation part further includes the generation of out-of-domain and in-domain data. Next, we will illustrate these four parts.

3.1 Data Filter

Following previous work (Li et al., 2019), we filter the training bilingual corpus with the following rules:

- Normalize punctuation with Moses scripts.
- Filter out the sentences longer than 100 words, or exceed 40 characters in a single word.

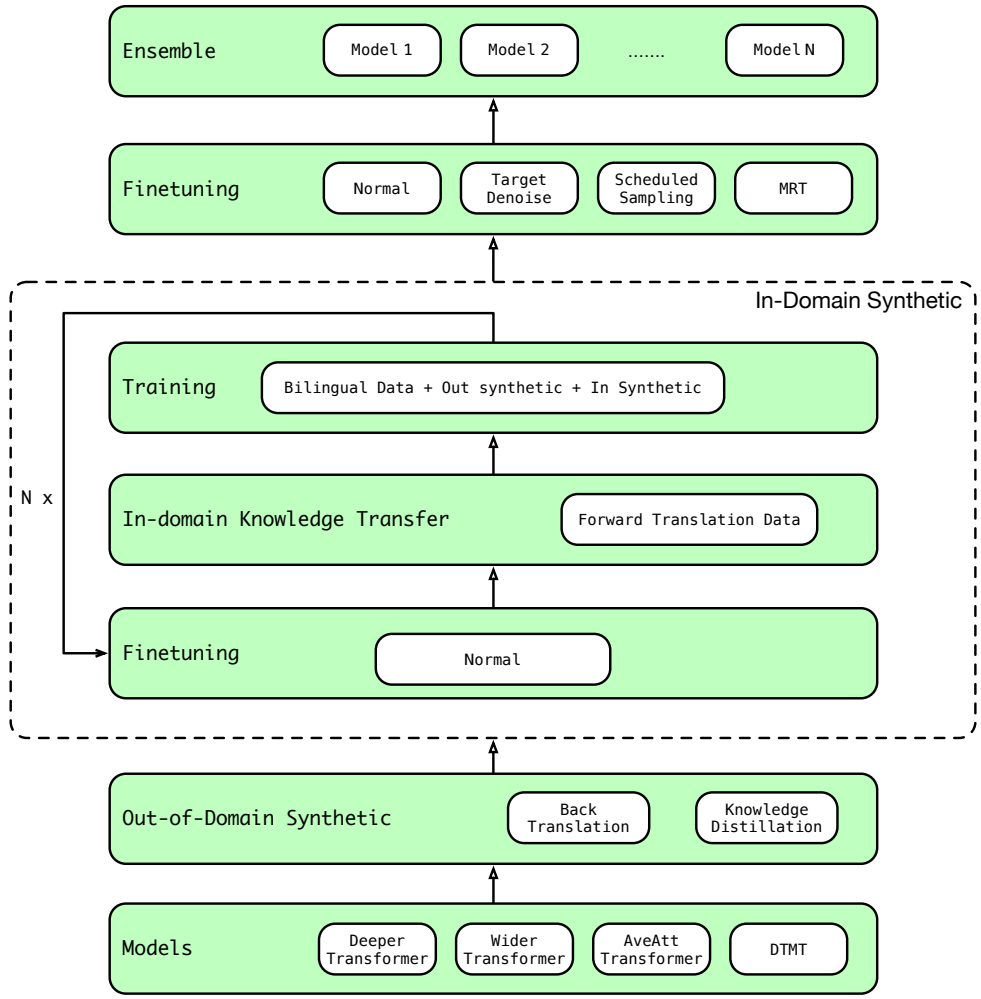


Figure 1: Architecture of WeChat NMT system. For simplicity, the data filtering module is ignored in the overview.

	NUM
Bilingual Data	20.7M
Chinese Monolingual Data	153.5M
English Monolingual Data	121.2M

Table 1: Statistics of all training data.

- Filter out the duplicated sentence pairs.
- The word ratio between the source and the target words must not exceed 1:4 or 4:1.

We also filter the monolingual corpus with the language model trained by the corresponding data of the bilingual training corpus.

In our experiments, the bilingual training data is a combination of News Commentary v15, Wiki Titles v2, WikiMatrix, CCMT and the UN corpus. The Chinese monolingual data includes News crawl, News Commentary, Common Crawl and Gigaword corpus. The English monolingual data includes News crawl, News discussions, Europarl v10, News Commentary, Common Crawl, Wiki dumps and the Gigaword corpus. After data filtering, statistics of all training data are shown in Table 1.

3.2 Out-of-Domain Synthetic Data Generation

Now, we describe our techniques for constructing both out-of-domain and in-domain synthetic data. The out-of-domain synthetic corpus is generated via both large-scale back-translation and knowledge distillation to enhance the models' performance for all domains. Then, we propose iterative in-domain knowledge transfer, which transfers in-domain knowledge to the huge monolingual corpus (i.e., Chinese), and builds our in-domain synthetic corpus. In the following sections, we elaborate above techniques in detail.

3.2.1 Large-scale Back-Translation

Back-translation is shown to be very effective to boost the performance of NMT models in both academic research (Hoang et al., 2018; Edunov et al., 2018) and previous years' WMT competitions (Deng et al., 2018; Sun et al., 2019; Ng et al., 2019; Xia et al., 2019). Following their work, we also train baseline English-to-Chinese models with the parallel data provided by WMT2020. Both the Left-to-Right Transformer (L2R) and the Right-to-Left Transformer (R2L) are used to translate the filtered monolingual English corpus combined with the English side of golden parallel bitext to Chinese. Then the generated Chinese text and the original English text are regarded as the source side and target side, respectively.

In practice, it costs us 7 days on 5 NVIDIA V100 GPU machines to generating all back-translated data.

3.2.2 Knowledge Distillation

Knowledge distillation (KD) is proven to be a powerful technique for NMT (Kim and Rush, 2016) to transfer knowledge from the teacher model to student models. In particular, we first use the teacher models to generate synthetic corpus in the forward direction (i.e., Chinese→English). Then, we use the generated corpus to train our student models.

In this work, with baseline Chinese→English models (i.e., L2R and R2L) as teacher models, we translate the Chinese sentences of the parallel corpus to English to form our synthetic KD dataset. The knowledge distillation costs about 2 days on 2 NVIDIA V100 GPU machines to generate all synthetic data.

3.3 Iterative In-domain Knowledge Transfer

Since in-domain finetuning demonstrates substantial BLEU improvements (Sun et al., 2019; Li et al., 2019), we speculate that the parallel data and the dev/test sets fall in different domains. Therefore, adapting our models to the target domain in advance will provide gains over the dev/test sets and give a better initialization point for in-domain finetuning. To this end, we use knowledge transfer to inject more in-domain information into our synthetic data.

In particular, we first use normal finetuning (see Section 3.5) to equip our models with in-domain knowledge. Then, we ensemble these models and use the ensemble model to translate the Chinese monolingual corpus into English. For our ensemble translator, we use 4 models with different architectures. Next, we pair original Chinese sentences with generated in-domain pseudo English sentences to form a pseudo parallel corpus. So far, the in-domain knowledge from ensembled models is transferred to the generated pseudo-parallel corpus. Finally, we retrain our model with both the in-domain pseudo-parallel and out-of-domain parallel data.

We refer to the above process as the in-domain knowledge transfer. In our experiments, we find that iteratively performing the in-domain knowledge transfer can further provide improvements (see Table 2). For each iteration, we replace the in-domain synthetic data and retrain our models, and it costs about 10 days on 8 NVIDIA V100 GPU machines. For the final submission, the knowledge transfer is conducted twice.

3.4 Data Augmentation

Aside from synthetic data generation, we also apply two data augmentation methods over our synthetic corpus. Firstly, adding synthetic/natural noises to training data is widely applied in the NLP fields (Li et al., 2017; Belinkov and Bisk, 2017; Cheng et al., 2019) to improve model robustness and enhance model performance. Therefore, we proposed to add token-level synthetic noises. Concretely, we perform random replace, random delete, and random permutation over our data. The probability of enabling each of the three operations is 0.1. We refer to this corrupted corpus as *Noisy* data.

Secondly, as illustrated in (Edunov et al., 2018), sampling generation over back-translation shows its potential in building robust NMT systems. Consequently, we investigate the performance of sampled synthetic data. For back-translated data, we replace beam search with sampling in its generation. For in-domain synthetic data, we replace the golden Chinese with the back sampled pseudo Chinese sentences. We refer to the data with sampling generation as *Sample* data.

As a special case, we refer to the without augmentation data as *Clean* data.

3.5 In-domain Finetuning

We train the model on large-scale out-of-domain data until convergence and then finetune it on small-scale in-domain data, which is widely used for domain adaption (Luong and Manning, 2015; Li et al., 2019). Specifically, we take Chinese→English test sets of WMT 17 and 18 as in-domain data, and filter out documents that are originally created in

English (Sun et al., 2019). We name above finetuning approach as *normal* finetuning. In all our finetuning experiments, we set the batch size to 4096, and finetune the model for around 400 steps[1] on the in-domain data.

Furthermore, the well-known problem of exposure bias in sequence-to-sequence generation becomes more serious under domain shift (Wang and Sennrich, 2020). To solve this issue, we further explore some *advanced* finetuning approaches and describe details in the following paragraphs.

Parallel Scheduled Sampling. We apply a two-pass decoding strategy for the Transformer decoder when finetuning, which is named as parallel scheduled sampling (Mihaylova and Martins, 2019; Duckworth et al., 2019). In the first pass, we obtain model predictions as a standard Transformer, and then mix the predicted sequence with the golden target sequence. In the second pass, we feed the above mixture of both golden and predicted tokens as decoder inputs for the final prediction. Thus the problem of the training-generation discrepancy is alleviated in the finetuning stage. According to our preliminary experiments, we set the proportion of predicted tokens in mixed tokens to 50%.

Target Denoising. In the training stage, the model never sees its own errors. Thus the model trained with teacher-forcing is prone to accumulated errors in testing (Ranzato et al., 2015). To mitigate this training-generation discrepancy, we add noisy perturbations into decoder inputs when finetuning. Thus the model becomes more robust to prediction errors by target denoising. Specifically, the finetuning data generator chooses 30% of sentence pairs to add noise, and keeps the remaining 70% of sentence pairs unchanged. For a chosen pair, we keep the source sentence unchanged, and replace the i-th token of the target sentence with (1) a random token of the current target sentence 15% of the time (2) the unchanged i-th token 85% of the time.

Minimum Risk Training. To further avoid the problem of exposure bias, we propose to use minimum risk training (Shen et al., 2016) in the finetuning stage, which directly optimizes the expected BLEU score instead of the Cross-Entropy loss, and naturally avoids exposure bias. Specifically, the objective is computed by,

$$R(\theta) = \sum_{s=1}^{S} \sum_{y \in S(x^{(s)})} Q(y|x^{(s)};\theta,\alpha)\Delta(y, y^{(s)}), \tag{1}$$

where $x^{(s)}$ and $y^{(s)}$ are two paired sentences. Δ denotes a risk function and $S(x^{(s)}) \in Y$ is a sampled subset of full search space. Then, the distribution Q is defined over space $S(x^{(s)})$,

$$Q(y|x^{(s)};\theta,\alpha) = \frac{P(y|x^{(s)};\theta)^{\alpha}}{\sum_{y' \in S(x^{(s)})} P(y'|x^{(s}};\theta)^{\alpha}}. \tag{2}$$

In practice, we use 4 candidates for each source sentence $x^{(s)}$. Although the paper claimed that sampling generates better candidates, we find that the beam search performs better in our extremely large Transformer model. The risk function we used is the 4-gram sentence-level BLEU (Chen and Cherry, 2014) and we tune the optimal α via grid search within $\{0.005, 0.05, 0.5, 1, 1.5, 2\}$. Each model is fine-tuned for a max of 1000 steps.

3.6 Ensemble

We split each training data into three shards among *Clean*, *Noisy* and *Sample* data respectively, which yields a total number of 9 shards. For each shard, we train seven varieties (two Deeper transformers, two Wider transformers, two AANs and one DTMT) with different model architecture. Then we apply four finetuning approaches on each model, thus the total number of models is quadrupled (about 200 models). For ensemble, it is difficult and inefficient to enumerate over all combinations of candidate models (e.g., grid search). Therefore a pruning strategy for model selection is necessary when ensemble. We try to greedily select the top-performing models for the ensemble. However, only slight improvement is obtained (less than 0.1 BLEU), as our models are too similar to each other after finetuning.

To further promote diversity among candidate models, we propose the self-bleu driven pruning strategy for *advanced* ensemble. Specifically, we take the translations of one model as hypotheses and translations of other models as references. Then we calculate the BLEU score for each model to evaluate its diversity among other models. Models with small BLEU scores are selected for ensemble, and vice versa. According to our experiments, we observe that (1) AAN and DTMT show

[1] According our experiments, finetuning with more steps will make the model easy to overfit on the small in-domain data.

SETTINGS	DEEPER	WIDER	AVEATT	DTMT
Baseline	26.24	26.35	26.17	26.08
+ Back Translation	29.64	29.70	29.48	28.88
+ *Finetune*	35.71	35.89	35.80	35.03
+ 1st In-domain Knowledge Transfer	38.14	38.22	38.21	37.98
+ *Finetune*	38.36	38.25	38.13	37.85
+ 2nd In-domain Knowledge Transfer	38.32	38.29	38.34	38.05
+ *Finetune*	38.49	38.31	38.38	38.12
+ *Advanced Finetune*	39.08	39.12	38.93	38.66
+ Normal Ensemble	39.19			
+ Advanced Ensemble⋆	**39.89**			

Table 2: Case-sensitive BLEU scores (%) on the Chinese→English *newstest2019*, where '⋆' denotes the submitted system. For each model architecture, we report the highest score among the three shards of clean data.

FINETUNING APPROACH	DEEPER	WIDER	AVEATT	DTMT
Normal	38.49	38.31	38.38	38.12
Parallel Scheduled Sampling	38.76	38.84	**38.93**	–
Target Denoising	38.88	38.92	38.63	**38.66**
Minimum Risk Training	**39.08**	**39.12**	38.78	38.45

Table 3: Case-sensitive BLEU scores (%) on the Chinese→English *newstest2019* for different finetuning approaches after the 2nd in-domain knowledge transfer. For each model architecture, we report the highest score among three shards of clean data and bold the best result among different finetuning approaches.

a clear difference with other architectures; (2) data sharding is effective to promote diversity, especially for models trained with *Clean* data; (3) different finetuning approaches cannot bring diversity for the same model. Under the guidance of self-bleu scores, our *advanced* ensemble models consist of 20 single models with differences in model architectures, data types, shards and finetuning approaches. As shown in Table 2, the *advanced* ensemble achieves absolute improvements over the normal ensemble (up to 0.7 BLEU improvements).

4 Experiments

4.1 Settings

All of our experiments are carried out on 15 machines with 8 NVIDIA V100 GPUs each of which has 32 GB memory. We use cased BLEU scores calculated with Moses[2] mteval-v13a.pl script as evaluation metric. *newstest2019* is used as the development set. For all experiments, we use LazyAdam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.998$ and $\epsilon = 10^{-9}$. The learning rate is set to 2.0 and decay with training steps. We use warmup step = 8000. We set beam size to 4 and alpha to 0.6 during decoding.

[2]http://www.statmt.org/moses/

4.2 Pre-processing and Post-processing

We segment the Chinese sentences with an in-house word segmentation tool. For English sentences, we successively apply punctuation normalization, tokenization and truecasing with the scripts provided in Moses. To enable open-vocabulary, we use byte pair encoding BPE (Sennrich et al., 2016b) with 32K operations for both Chinese and English sides.

For the post-processing, we apply de-truecaseing and de-tokenizing on the English translations with the scripts provided in Moses.

4.3 Main Results

Table 2 shows that the translation quality is largely improved with proposed techniques. We observe a solid improvement of 2.8∼3.4 BLEU for the baseline system after back translation. In-domain finetuning yields substantial improvements among all model architectures, which are 6.07∼6.32 BLEU. The finetuned Transformer models achieve about 35.89 BLEU scores, and the DTMT achieves a 35.03 BLEU score. These findings demonstrate that the domain of training corpus is apart from the target domain, and hence domain adaptation has great potential in improving model performance in the target domain.

As described in Section 3.3, we inject the in-domain knowledge into our monolingual corpus.

Two In-domain knowledge transfers provide another up to 3.02 BLEU score gain (i.e., from about 35.03 to 38.05). The in-domain knowledge transfer brings more improvement compared with the normal finetuned models. Besides, we find that models further finetuned after in-domain transfer performs slightly better (about 0.1 BLEU). The improvement suggests that although in-domain transfer has already provided plenty of in-domain knowledge, it still has room for in-domain finetuning. We further apply *advanced* finetuning techniques to our models, as described in Section 3.5. The advanced finetuning further brings about 0.81 BLEU score gains, and we obtain our best single model with 39.12 BLEU scores.

In our preliminary ensemble experiments, we combine some top-performing models at each decoding step, but only achieve slight improvement over single models (about 0.1 BLEU). With our *advanced* ensemble strategies in section 3.6, further improvements are achieved over the normal ensemble (0.7 BLEU). As a result, our WMT 2020 Chinese→English submission achieves a cased BLEU score of 36.9 on *newstest2020*, which is the highest among all submissions.

4.4 Effects of Advanced Finetuning Approaches

In this section, we describe our experiments on advanced finetuning. Here we take clean models as examples, but models trained with noisy data and sampled data show similar trends.

As shown in Table 3, all three advanced finetuning methods significantly outperform normal finetuning. For Wider and Deeper Transformers, Minimum Risk Training provides the highest BLEU gain, which is 0.81. For the Average Attention Transformer, Parallel Schedule Sampling improves the model performance from 38.38 to 38.93. For the DTMT model, Target Denoising performs the best, improving from 38.12 to 38.66. These findings are in line with the conclusion of Wang and Sennrich (2020) that links exposure bias with domain shift. For each type of model, we only keep the best-performing finetuned one for the final model ensemble.

5 Conclusion

In this paper, we introduce the system WeChat submitted for the WMT 2020 shared task on Chinese→English news translation. Our system is based on the Transformer (Vaswani et al., 2017) with different variants and the DTMT (Meng and Zhang, 2019) architecture. Data selection, several effective synthetic data generation approaches (i.e., back-translation, knowledge distillation, and iterative in-domain knowledge transfer), advanced finetuning approaches (i.e., parallel scheduled sampling, target denoising, and minimum risk training) and self-bleu based model ensemble are employed and proven effective in our experiments. Our constrained Chinese→English system achieved 36.9 case-sensitive BLEU score which is the highest among all submissions.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Yonatan Belinkov and Yonatan Bisk. 2017. Synthetic and natural noise both break neural machine translation. *arXiv preprint arXiv:1711.02173*.

Boxing Chen and Colin Cherry. 2014. A systematic comparison of smoothing techniques for sentence-level bleu. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 362–367.

Yong Cheng, Lu Jiang, and Wolfgang Macherey. 2019. Robust neural machine translation with doubly adversarial inputs. *arXiv preprint arXiv:1906.02443*.

Yongchao Deng, Shanbo Cheng, Jun Lu, Kai Song, Jingang Wang, Shenglan Wu, Liang Yao, Guchun Zhang, Haibo Zhang, Pei Zhang, et al. 2018. Alibaba's neural machine translation systems for wmt18. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 368–376.

Daniel Duckworth, Arvind Neelakantan, Ben Goodrich, Lukasz Kaiser, and Samy Bengio. 2019. Parallel scheduled sampling. *arXiv preprint arXiv:1906.04331*.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. *arXiv preprint arXiv:1808.09381*.

Vu Cong Duy Hoang, Philipp Koehn, Gholamreza Haffari, and Trevor Cohn. 2018. Iterative back-translation for neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 18–24.

Yoon Kim and Alexander M Rush. 2016. Sequence-level knowledge distillation. *arXiv preprint arXiv:1606.07947*.

Bei Li, Yinqiao Li, Chen Xu, Ye Lin, Jiqiang Liu, Hui Liu, Ziyang Wang, Yuhao Zhang, Nuo Xu, Zeyang Wang, Kai Feng, Hexuan Chen, Tengbo Liu, Yanyang Li, Qiang Wang, Tong Xiao, and Jingbo Zhu. 2019. The NiuTrans machine translation systems for WMT19. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 257–266, Florence, Italy. Association for Computational Linguistics.

Jiwei Li, Will Monroe, Tianlin Shi, Sébastien Jean, Alan Ritter, and Dan Jurafsky. 2017. Adversarial learning for neural dialogue generation. *arXiv preprint arXiv:1701.06547*.

Yunlong Liang, Fandong Meng, Jinchao Zhang, Jinan Xu, Yufeng Chen, and Jie Zhou. 2019. A novel aspect-guided deep transition model for aspect based sentiment analysis. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5569–5580, Hong Kong, China. Association for Computational Linguistics.

Yijin Liu, Fandong Meng, Jinchao Zhang, Jinan Xu, Yufeng Chen, and Jie Zhou. 2019. GCDT: A global context enhanced deep transition architecture for sequence labeling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2431–2441, Florence, Italy. Association for Computational Linguistics.

Minh-Thang Luong and Christopher D Manning. 2015. Stanford neural machine translation systems for spoken language domains. In *Proceedings of the International Workshop on Spoken Language Translation*, pages 76–79.

Fandong Meng and Jinchao Zhang. 2019. DTMT: A novel deep transition architecture for neural machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 224–231.

Tsvetomila Mihaylova and André F. T. Martins. 2019. Scheduled sampling for transformers. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 351–356, Florence, Italy. Association for Computational Linguistics.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook fair's wmt19 news translation task submission. *arXiv preprint arXiv:1907.06616*.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2015. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Shiqi Shen, Yong Cheng, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. 2016. Minimum risk training for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1683–1692, Berlin, Germany. Association for Computational Linguistics.

Meng Sun, Bojian Jiang, Hao Xiong, Zhongjun He, Hua Wu, and Haifeng Wang. 2019. Baidu neural machine translation systems for WMT19. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 374–381, Florence, Italy. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Chaojun Wang and Rico Sennrich. 2020. On exposure bias, hallucination and domain shift in neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3544–3552, Online. Association for Computational Linguistics.

Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F. Wong, and Lidia S. Chao. 2019. Learning deep transformer models for machine translation. pages 1810–1822.

Lijun Wu, Yiren Wang, Yingce Xia, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. 2019. Exploiting monolingual data at scale for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4207–4216, Hong Kong, China. Association for Computational Linguistics.

Yingce Xia, Xu Tan, Fei Tian, Fei Gao, Weicong Chen, Yang Fan, Linyuan Gong, Yichong Leng, Renqian Luo, Yiren Wang, et al. 2019. Microsoft research asia's systems for wmt19. *arXiv preprint arXiv:1911.06191*.

Biao Zhang, Deyi Xiong, and Jinsong Su. 2018. Accelerating neural transformer via an average attention

network. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1789–1798, Melbourne, Australia. Association for Computational Linguistics.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A benchmarking platform for text generation models. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 1097–1100.

PROMT Systems for WMT 2020 Shared News Translation Task

Alexander Molchanov
PROMT LLC
17E Uralskaya str. building 3, 199155,
St. Petersburg, Russia
Alexander.Molchanov@promt.ru

Abstract

This paper describes the PROMT submissions for the WMT 2020 Shared News Translation Task. This year we participated in four language pairs and six directions: English-Russian, Russian-English, English-German, German-English, Polish-English and Czech-English. All our submissions are MarianNMT-based neural systems. We use more data compared to last year and update our back-translations with better models from the previous year. We show competitive results in terms of BLEU in most directions.

1 Introduction

This paper provides an overview of the PROMT submissions for the WMT 2020 Shared News Translation Task. This year we participate with neural MT systems for the third time. We participate in four language pairs and six directions. We describe our data preparation pipelines, models training setups and present the results on the newstest sets.

The paper is organized as follows: Section 2 is a brief overview of the submitted systems. Section 3 describes the data preparation, preprocessing and statistics in detail. Section 4 provides a detailed description of the systems. In Section 5 we present and discuss the results. Section 6 concludes the paper.

2 Systems overview

We submitted six systems based on the `MarianNMT` (Junczys-Dowmunt et al., 2018) toolkit: English-Russian, Russian-English, English-German, German-English, Polish-English and Czech-English. All systems are unconstrained (we use the allowed data, private data and publicly available unconstrained data like OpenSubtitles). The English-German and German-English systems have the same basic architecture. The English-Russian and Russian-English systems are slightly different as we use separate vocabularies. The Polish-English system was trained jointly in both directions. The Czech-English is a multilingual system trained to translate from Croatian, Serbian, Slovak and Czech to English.

3 Data

We use all data provided by the WMT organizers, private in-house parallel data and other publicly available data, mainly from the OPUS website (Tiedemann, 2012). The human parallel data for the German-English system is exactly the same as for the English-German system, the two systems only have different synthetic back-translated data. This also applies to the English-Russian and Russian-English systems.

We use the Tatoeba sets as our validation sets and the newstest2019 is our test set. The reason why we choose the Tatoeba corpus for validation is that we aim at building general-domain (and not just news-domain) models. Besides, the Tatoeba corpus is available for many language pairs beyond the scope of the WMT Translation Task.

We only do fine-tuning for the Czech-English system. This will be described in detail in Section 3.4 below.

3.1 Data filtering

There are several stages in our data filtering pipeline. The statistics for the final training data are shown in Table 1. Note that for the multilingual Czech|Croatian|Serbian|Slovak-English system the table provides statistics only for the Czech-English part. The size of the filtered versions of the Croatian, Serbian and Slovak parts

Proceedings of the 5th Conference on Machine Translation (WMT), pages 248–253

are 21.8M, 21.7M and 14M parallel sentences respectively (more than 95% of the Serbian data is OpenSubtitles).

Basic filtering

This includes some simple length-based and source-target length ratio-based heuristics, removing tags, lines with low amount of alphabetic symbols etc. We also remove lines which appear to be emails or web-addresses. In addition, we remove lines with rare words from the Bookshop and the OpenSubtitles corpora (using frequency lists built on large monolingual corpora including all monolingual data from WMT, private data and Wikipedia dumps).

Deduplication

We remove duplicate translations and keep only the most frequent translation for the source sentence if it repeats more than two times. This procedure is applied to some corpora, e.g. OpenSubtitles and MultiUN which contain a lot of various (and often incorrect) translations for common phrases. For example, the English phrase '*No.*' is encountered almost 100k times in the source side of the English-Russian OpenSubtitles corpus. It has more than 78k unique translations, second most popular among which is '*Да.*' ('*Yes.*' in Russian).

Language detection

The algorithm is a fairly simple ensemble of three tools: `pycld2` [1], `langid` (Lui and Baldwin, 2012), `langdetect` [2].

Parallel data filtering with NMT and language models

We apply this step to all data. Last year we used our own algorithm based on `Hunalign` (Varga et al., 2005) and our inhouse classifier to identify and discard unparallel sentence pairs. This year we use a different approach. We score parallel data with NMT models in both directions. We also score the source and target sides of the data with statistical language models built on large sets of what we assume to be good-quality data (basically, the newscrawl data from the statmt.org website). The scores are normalized by sentence length and summed up. We also apply weights (from 0.1 to 0.3 depending on the corpus type) to the statistical LMs scores as we mostly rely on the scores produced by the NMT models. The data is then sorted according to the final scores, and we select a subset of the data according to a certain threshold set individually for different corpora by our linguists.

	German-English		Russian-English		Polish-English		Czech-English	
	#sent	#tokens EN	#sent	#tokens EN	#sent	#tokens EN	#sent	#tokens EN
WMT	26.6	580.1	27.3	690.9	10.3	183.2	11.4	147.8
OPUS	23.8	475.9	8.3	74.9	26.8	283.7	29.1	263.8
Private	7.5	100.4	25.5	428.2	0.3	3.7	0.4	5.1
Total	57.9	1156.4	61.1	1194.0	37.4	470.6	40.9	416.7

Table 1: Statistics for the filtered parallel data in millions of sentences (#sent) and tokens (#tokens) for four language pairs. WMT stands for the data available for the News Task on the statmt.org/wmt20 website; OPUS is the data from the OPUS website apart from the data available for the News Task; Private stands for private company data.

3.2 Data preprocessing

BPE

Same as last year, we use the `OpenNMT` toolkit (Klein et al., 2017) version of byte pair encoding (BPE) (Sennrich et al., 2016b) to encode our data to subword units. The BPE merge operations are learnt in case-insensitive mode. Case-insensitive BPE model is very useful when dealing with noisy data (like, for example, OpenSubtitles where uppercase is often used to communicate emphasis) or legal and financial data where specific terms are written in title case or uppercase. News headlines are also often written in title case or uppercase.

The OpenNMT preprocessor handles case as a feature assigned to each token. As MarianNMT

[1] https://pypi.org/project/pycld2/
[2] https://pypi.org/project/langdetect/

does not support features yet, we perform a 'trick' similar to the one described in (Tamchyna et al., 2017): instead of using a feature we insert special tokens <C> and <U> after sequences in title case or uppercase. For example, a source sentence

World Championships 2017: Neil Black praises Scottish members of Team GB

is converted to

world <C> championships <C> 2017 : neil <C> black <C> pra@@ ises scottish <C> members of team <C> gb <U>

We do not use truecaser in our pipeline as it is redundant. All data is tokenized using the `Moses` toolkit (Koehn et al., 2007) tokenizer with aggressive tokenization, then the OpenNMT BPE-splitter is applied, after that we convert the case feature to separate tokens.

General tendency for our models this year is to build smaller BPE models.

English-Russian and Russian-English

Same as last year (Molchanov, 2019), we train the models with separate vocabularies due to the Cyrillic nature of Russian alphabet. Therefore we build separate BPE models for source and target, but with less merge operations (16k for English and 32k for Russian) compared to last year (35k and 45k respectively).

English-German and German-English

We train a joint BPE model for the English-German pair with 16k merge operations. We use a shared vocabulary and tie all embeddings for all translation models with joint BPE.

Polish-English

We train a joint BPE model for the Polish-English pair with 12k merge operations.

Czech-English

As was mentioned earlier, our Czech-English model is a multilingual model trained to translate from Czech, Croatian, Serbian and Slovak into English. Therefore we train a joint BPE model for all five languages with 24k merge operations. As part of Serbian data is in Cyrillic alphabet, we transliterate it into Latin using an inhouse transliteration tool.

3.3 Synthetic data

There are two types of additional synthetic training data described in detail below. The final size of the training data for the submitted systems is roughly 4 times the total size of the filtered data in Table 1 Table 1 for each language pair.

Both types of synthetic data are used for training all submitted systems. We also tag all synthetic data following (Caswell et al., 2019), i.e., insert a special token <*bt*> at the beginning of each source line for back-translations etc.

Back-translated data

Back-translations (Sennrich et al., 2016a) are a common way to improve NMT models quality. As we aim at building general-domain models, we use data from Wikipedia dumps and news from statmt.org. We shuffle the Wikipedia data and randomly select a subset of appropriate size. The selected Wikipedia subset and the news subset are roughly equal in size. The size of the whole corpus used for back-translation is approximately equivalent to the size of human training data.

For the English-Russian pair we use our last year's English-Russian model to obtain back-translations for the Russian-English model. Then we train the Russian-English model and use it to obtain back-translations for the final English-Russian model.

We also obtain back-translations for the German-English pair using our last year's models.

For the Polish-English and Czech-English pairs we build intermediate models using all available data excluding OpenSubtitles and Paracrawl.

We score our back-translations with the opposite-direction NMT models to discard obviously bad translations.

Replicated data with unknown words

We apply the technique described in (Pinnis et al., 2017) to create a synthetic parallel corpus. The procedure includes the following steps: first, we perform word-alignment of our initial parallel training corpus using the fast-align tool (Dyer et al., 2013). Then, we randomly replace from one to three unambiguously (one-to-one) aligned tokens in both source and target parallel sentences with the special <UNK> placeholder. The same pipeline is applied to both the initial and back-translated data. We train our models to reproduce the <UNK> placeholder in various contexts and

use this feature for handling named entities as described in Section 4.1 below.

3.4 Data for fine-tuning

We only do fine-tuning for the Czech-English system. The model is tuned on available parallel Czech-English data mixed with back-translations of the English news 2017-2019 from statmt.org. We use the newstest2017-2019 as our devset.

4 Systems architecture

This section describes the trained systems in detail. We train transformer (Vaswani et al., 2017) models for all submitted systems. We use the recipe available at the MarianNMT website[3]. The system configuration, hyperparameters and training steps follow those in the recipe.

We use the transformer-big configuration for the English-Russian model.

We train single models for all directions.

We use the beam of size 6 and the `--normalize` parameter is set to 0.6.

4.1 Handling named entities

We preserve several types of named entities (NEs): numbers, emails, alphanumeric sequencies etc. in the following way. First, we produce the baseline NMT translation without any processing. Then we validate the translation of NEs by comparing the system's output to the source sentence. The validation is simple: we search for the corresponding strings (numbers, emails etc.) in the system's output. If some of the NEs are not translated or are translated incorrectly, we replace the entities with the <UNK> placeholder in the source sentence and translate the sentence again allowing the decoder to generate unknown words in the output. Finally, we substitute the <UNK> placeholders in the output with their initial value. If the number of the <UNK> placeholders in the NMT system's output is not equal to the number of the placeholders in the source sentence, we fall back to the baseline NMT translation without NEs processing. We do not do any specific processing for proper names.

5 Results and discussion

In this section we present the BLEU (Papineni et al., 2002) scores for our systems on two test sets and the analysis of the results.

The scores are presented in Table 2. Calculation is done using the `multi-bleu-detok.perl` script from the `Moses` toolkit.

We significantly outperform our last year's submissions for the News Task.

Fine-tuning for the Czech-English system does not give us significant improvements in terms of BLEU. This may be because we didn't perform any data selection this year.

System	newstest2019	newstest2020
English-Russian		
Model2019	29.5	21.7
Model2020	**32.3**	**23.3**
Russian-English		
Model2019	37.2	33.6
Model2020	**42.3**	**38.2**
Polish-English		
Model2020	-	**31.3**
English-German		
Model2019	38.2	29.8
Model2020	**40.7**	**31.9**
German-English		
Model2019	32.4	34.9
Model2020	**39.4**	**39.6**
Czech-English		
Model2020	-	25.1
Model2020 tuned	-	**25.6**

Table 2: Results for different systems and directions. The submitted systems are marked in bold. Model2019 stands for our last year's submitted systems which we consider the baseline.

We are among the top 10 systems in the English<>Russian directions, however, we are substantially behind the top systems in other directions in terms of BLEU. We see two reasons for that. First of all, we pay much more attention to our Russian systems, thus, our last year's Russian systems had already undergone several iterations of updated backtranslations and retraining and can be considered strong baselines. Second, we possess much more private high-quality data for the English-Russian pair compared to other language pairs.

[3] https://github.com/marian-nmt/marian-examples/tree/master/wmt2017-transformer

6 Conclusions and Future work

In this paper we have described our submissions for the WMT 2020 Shared News Translation Task. Overall we have made six submissions in four language pairs: English-Russian, English-German, Polish-English and Czech-English.

We have documented the methodology used to prepare the training data, system training set-ups, the pipeline for handling NEs.

We show competitive results in most directions.

In future we plan to experiment once again with a shared vocabulary for the English-Russian models applying transliteration to the source side.

References

Isaac Caswell, Ciprian Chelba, and David Grangier. 2019. Tagged Back-Translation. In *Proceedings of the Fourth Conference on Machine Translation*, pages 53–63, Florence, Italy.

Anna Currey, Antonio Valerio Miceli Barone, and Kenneth Heafield. 2017. Copied monolingual data improves low-resource neural machine translation. In *Proceedings of the Second Conference on Machine Translation*, pages 148–156, Copenhagen, Denmark.

Chris Dyer, Victor Chahuneau, and Noah A Smith. 2013. A Simple, Fast, and Effective Reparameterization of IBM Model 2. In *Proceedings of NAACL HLT 2013*, pages 644–648, Atlanta, USA.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast Neural Machine Translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, Alexander M. Rush. 2017. OpenNMT: Open-Source Toolkit for Neural Machine Translation. *Computing Research Repository*, arXiv:1701.02810. Version 2.

Marco Lui and Timothy Baldwin. 2012. langid.py: An Off-the-shelf Language Identification Tool. In *Proceedings of ACL 2012, System Demonstrations*, pages 25–30, Jeju, Republic of Korea.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions, ACL 07*, pages 177–180, Stroudsburg, PA, USA.

Alexander Molchanov. 2019. PROMT Systems for WMT 2019 Shared Translation Task. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 302–307, Florence, Italy.

Kishore Papineni, Salim Roukos, Todd Ward, and WeiJing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL 02)*, pages 311–318, Philadelphia, PA, USA.

Marcis Pinnis, Rihards Krišlauks, Daiga Deksne, and Toms Miks. 2017. Neural Machine Translation for Morphologically Rich Languages with Improved Sub-word Units and Synthetic Data. In *Proceedings of the 20th International Conference of Text, Speech and Dialogue (TSD2017)*, pages 237–245, Prague, Czechia.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving Neural Machine Translation Models with Monolingual Data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*, Berlin, Germany.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. 2016b. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*, pages 35–40, Berlin, Germany.

Michel Simard, Cyril Goutte, and Pierre Isabelle. 2007. Statistical phrase-based post-editing. In *Proceedings of The North American Chapter of the 342 Association for Computational Linguistics Conference (NAACL-07)*, pages 508–515, Rochester, NY, USA.

Aleš Tamchyna, Marion Weller-Di Marco and Alexander Fraser. 2017. Modeling Target-Side Inflection in Neural Machine Translation. In *Proceedings of the Second Conference on Machine Translation*, pages 32–42, Copenhagen, Denmark.

Jorg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, pages 2214–2218, Istanbul, Turkey.

Dániel Varga, László Németh, Péter Halácsy, András Kornai, Viktor Trón, Viktor Nagy. 2005. Parallel corpora for medium density languages. In *Proceedings of the RANLP 2005*, pages 590–596, Borovets, Bulgaria.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS 2017)*, Long Beach, CA, USA.

eTranslation's Submissions to the WMT 2020 News Translation Task

Csaba Oravecz† **Katina Bontcheva**† **László Tihanyi**† **David Kolovratnik**†
Bhavani Bhaskar† **Adrien Lardilleux**† **Szymon Klocek**⋆ **Andreas Eisele**⋆
DG Translation – DG CNECT, European Commission
†firstname.lastname@ext.ec.europa.eu
⋆firstname.lastname@ec.europa.eu

Abstract

The paper describes the submissions of the eTranslation team to the WMT 2020 news translation shared task. Leveraging the experience from the team's participation last year we developed systems for 5 language pairs with various strategies. Compared to last year, for some language pairs we dedicated a lot more resources to training, and tried to follow standard best practices to build competitive systems which can achieve good results in the rankings. By using deep and complex architectures we sacrificed direct re-usability of our systems in production environments but evaluation showed that this approach could result in better models that significantly outperform baseline architectures. We submitted two systems to the zero shot robustness task. These submissions are described briefly in this paper as well.

1 Introduction

The European Commission's eTranslation project[1], a building block of the Connecting Europe Facility (CEF), has been set up to help European and national public administrations exchange information across language barriers in the EU. More details about the project can be found in (Oravecz et al., 2019). Our participation in last year's WMT shared task marked an important step towards opening the service to the coverage of additional, non-EU languages and to domains beyond the formal language of EU institutions. Due to the encouragement and insights we received from WMT 2019, a complete set of general domain MT engines has meanwhile been implemented and incorporated into the eTranslation service.

This year the team participated in the news translation shared task with five different language pairs: English → German, Japanese → English, English → Polish, Russian → English and English → Czech. The varying performance of these systems reflects the amount of resources dedicated to their developments.

2 Data Preparation

This section briefly describes the data sets, the selection, and filtering methods applied to the provided parallel and monolingual data in order to increase the quality of trained models. We primarily focused on constrained submissions, but due to the low quality of our first En→Pl models trained only on the constrained data set we switched to the unconstrained scenario and chose to submit only the unconstrained En→Pl system (see Section 4.3).

2.1 Data Selection and Filtering

In general, we made use of all provided original parallel (OP) data to build baseline models for reference or back-translation. Some brief experiments were made with the exclusion of one or the other data set. However, the best baseline models were trained when we used all OP data (except for the UN Parallel Corpus for Ru→En, which, like last year, did not improve the results). This year, where we used it, we did not apply any advanced filtering technique to ParaCrawl (except for JParaCrawl for Ja→En) either, the 5.1 version proved to be usable without further complex processing.

The domain distribution of the data sets was not uniform across language pairs, which had some influence on the workflows we applied to specific language pairs but the basic procedure of data cleaning was similar in all cases.

As a general clean-up, we performed the following steps on the parallel data[2]:

- language identification with FastText[3] (Joulin et al., 2016),

[1] https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITAL/eTranslation

[2] For Japanese, these steps were not used.

[3] https://fasttext.cc/docs/en/language-identification.html

Proceedings of the 5th Conference on Machine Translation (WMT), pages 254–261
Online, November 19–20, 2020.

Data set	En→De	Ja→En	En→Pl	Ru→En	En→Cs
Europarl v10	1.80M	–	0.62M	–	0.62M
Common Crawl	2.18M	–	–	0.78M	0.11M
News Commentary v15	0.36M	1.74k	–	0.30M	0.25M
Rapid Corpus	1.12M	–	0.25M	–	0.28M
Wiki Titles v2	1.30M	0.59M	0.49M	0.05M	0.32M
Yandex	–	–	–	1.00M	–
(J)ParaCrawl	34.2M	8.63M	6.18M	4.25M	4.90M
WikiMatrix	5.47M	0.81M	0.55M	3.40M	1.92M
CzEng 2.0	–	–	–	–	41.6M
TED Talks	–	0.23M	–	–	–
Subtitle Corpus	–	2.80M	–	–	–
Kyoto Free	–	0.43M	–	–	–
Total:	46.43M	13.49M	8.04M	9.78M	50.0M

Table 1: Number of segments in the filtered parallel data used for baseline models.

- segment deduplication with masked numerals[4],
- deletion of segments where source/target token ratio exceeds 1:3 (or 3:1),
- deletion of segments longer than 100-150 tokens (depending on language pair),
- exclusion of segments without a minimum number of alphabetic characters.

The above steps led to an average reduction of about 10% of the training data.

We applied language specific filtering in Ja→En to exclude segments which contained non-Latin (Greek) or non-CJK character ranges, and in En→Cs we added a sentence segmentation step using Tikal[5] to break up a large number of raw segments merging several sentences. In the En→Pl and Ru→En data sets, we filtered out segments with more than 8 or mismatched numeric tokens, and deleted segments filled with excessive punctuation marks. The number of segments in the base filtered data is shown in Table 1.

In the language pairs where we used monolingual data to build language models or create synthetic parallel text, we generally selected recent target language News Crawl data sets. For En→Pl, the 1.32B segments of the Polish Common Crawl were ranked with a language model built on the News Crawl data, and the top 2.15M segments were used for back-translation. In the non-Japanese back-translation data we performed some additional filtering: we set a threshold on the maximum length of a token (40-100) and the minimum ratio of letters to digits in a segment (4), filtered out segments with scrambled tokens (2019 German News Crawl) or token (bigram) repetitions (En→Cs).

Depending on data availability we needed different ways of creating development and test data sets. For En→De and En→Cs, we used the 2018 test set as validation set in the trainings and the 2019 test set as the test set to evaluate the trained models. We did not specifically make a source original extraction from these data sets; the 2019 test set already contained only source original segments and the 2018 set was only used for early stopping of the training (see Section 3.2.2 for the use of source original data sets in the trainings).

For Ru→En, we used the 2018 and 2019 test sets for testing and 2500 segments randomly selected from the combined 2012-2017 test sets. For En→Pl, due to data sparsity, we used 500 segments of the 2020 dev set for testing and the rest for validation. For Ja→En, the provided development set was used to test the models during development, while a random subset of 3000 segments from OP was extracted to serve as validation set.

2.2 Pre- and Postprocessing

Similarly to our last year's submissions (Oravecz et al., 2019), in the default workflows, we generally did not apply the standard pre- and postprocessing steps of truecasing, or (de)tokenization, these

[4]We deleted duplicate segments regardless of differences in numerals.

[5]https://okapiframework.org/

did not have a noticeable effect on most of the results. We simply used SentencePiece (Kudo, 2018), which allows raw text input/output within the Marian toolkit (Junczys-Dowmunt et al., 2018)[6] in the experiments. For certain language pairs, however, some tailored processing steps were applied and tested. These are described in detail in the language pair specific result sections.

3 Trainings

Our access to computing resources is not unlimited. Therefore, we did not have much room for large scale experiments with either a wide range of scenarios or extensive tuning of hyperparameters. Nevertheless, as opposed to last year, where we decided to stick only to simple setups and training procedures, this year we tried more complex models and utilized significantly more data where it was possible. In all experiments we used Marian, which is the core tool of our standard NMT framework in the eTranslation service. All trainings were run as multi-GPU trainings on 2 or 4 NVIDIA V100 GPUs with 16GB RAM. Base transformers were typically trained for 7 epochs for high resource and 11 epochs for lower resource language pairs, whereas big transfomers were generally trained for 12 epochs for high and 30 epochs for lower resource.

3.1 NMT Models

We trained base transformer models (Vaswani et al., 2017) in all language pairs for the first baseline models and for models used for back-translation to gain efficiency in back-translating large amounts of target monolingual data. To build the more competitive systems we switched to big transformer architectures; this in some cases led to significant improvements but at the same time the rise in computing costs was also substantial. This year we also built 2–4 member ensembles from big transformers for high resource language pairs; again a high cost for a relatively smaller scale improvement. For most of the hyperparameters we used the default settings for the base transformer architecture in Marian[7] with dynamic batching and tying all embeddings. To save time and resources, we stopped the trainings if sentence-wise normalized cross-entropy on the validation set did not improve in 5 consecutive validation steps. In the big transformer experiments, following recommended settings for Marian, we doubled the filter size and the number of heads, decreased the learning rate from 0.0003 to 0.0002 and halved the update value for `--lr-warmup` and `--lr-decay-inv-sqrt`.

For En→De and En→Cs we set a 36k joint SentencePiece vocabulary, which seems to be more or less in the standard range nowadays. We had some previous experiments with other vocabulary sizes but with no improvement. Ja→En models were trained with a 32k vocabulary size, En→Pl with 32k, and Ru→En with 30k.

3.2 Improving Baseline Models

This section describes the methods we applied to improve baseline models, such as building additional synthetic data sets with back-translation (Sennrich et al., 2016), using original parallel or development data (where available) to continue the training of already converged models and building ensembles of deep models originally trained from different seeds. Evaluation scores are reported in Section 4.

3.2.1 Synthetic Data

Back-translation (BT) is a standard data augmentation technique in neural machine translation, but one which brings another set of tunable parameters in the search for best settings as far as the optimal amount of synthetic data, ratio of bitext to back-translation data or methods to generate the synthetic source are concerned (Edunov et al., 2018; Hoang et al., 2018). Tagged back-translation (Caswell et al., 2019) has recently been proposed as a simple alternative to noising techniques, arguing that it is the indication of the data being synthetic that is relevant for the model. This has been justified in our experiments as well, therefore we used this technique in all workflows for all language pairs.

In the En→De system, we ran various experiments with small amounts of BT data from the 2019 News Crawl (10M, 20M, 50M), which gave some improvement in the base architectures. However, for the deeper models we back-translated 116M[8] 2016, 2017 and 2019 News Crawl segments and used it as tagged synthetic data in the trainings (with segments longer than 75 tokens filtered

[6] We used default settings for Marian's built-in SentencePiece: unigram model, built-in normalization and no subword regularization.

[7] See eg. `https://github.com/marian-nmt/marian-examples/tree/master/transformer`.

[8] From 170M after the filtering.

out). As suggested by Ng et al. (2019) and Junczys-Dowmunt (2019), we upsampled the original parallel data to a 1:1 ratio.[9] This setup was a one shot configuration, we had no time and resources to experiment with using more BT data or other OP-BT combinations. In En→Cs we followed a similar procedure of back-translating recent News Crawl data and upsampling the OP data to keep the balance of the two types of data sets.

For Ja→En, we tried to use only News Crawl or use it together with the News Discussions monolingual data. Both setups gave similar results in the end. In Ru→En, we first experimented with the BT data provided by the University of Edinburgh but this was not beneficial so we decided to use only translations produced by our own BT systems. We translated 100M of the monolingual English data (50M News Crawl (2017-2019) and 50M News Discussions (2018–2019)), and filtered it down with LMs to 50.4M.

For En→Pl, we translated all of the available Polish News Crawl (3.79M) as well as 2.15M of the Polish Common Crawl (cf. Section 2.1). They were subsequently filtered down to 3.7M and 1.97M.

3.2.2 Continued Trainings

This year we experimented with a two stage continued training process as a possible direction to improve performance as domain adaptation (Luong and Manning, 2015). For En→De, we built a transformer language model from the 2016, 2017 and 2019 filtered News Crawl data set (116M segments) and scored the German side of the original parallel data. The scores created a ranking of OP data from which we took the top 20M[10] to continue training of OP+BT trained models (as suggested by Junczys-Dowmunt (2019)) until the BLEU score on the test set increased (typically 2 epochs with an increase of 1 point). In the second stage, we used the 2008–2018 development sets (32.5k segments) in the experiments and for the final submission we extended it with the 2019 test set. We trained 4 epochs on this set and then for additional 2 epochs we switched to a source original subset (14.5k) to reach the highest BLEU score. This second stage yielded a much smaller improvement than last year. However, this year the starting models were more powerful already. Fine tuning on the development set worked much better for Ja→En, where we achieved more than 2 points BLEU score (Table 3) increase on the best performing engine by continuing the training until the first stall (20 epochs). The same procedure, however, did not give any improvement for En→Cs.

3.2.3 Ensembles

For the En→De final submissions, we set up a 4 big transformer ensemble trained with the same (best) configuration and workflow but with different seeds. As reported in Section 4.1, this system achieved the highest score and was submitted as primary. In Ja→En, a two model ensemble did not yield any improvement so it was not submitted, in En→Cs, a two model ensemble was submitted because it outperformed a three model one on the development set. The Ru→En and the En→Pl systems submitted were 3 model big transformer ensembles the latter with only a minimal increase in performance compared to the single models (cf. Section 4.3).

3.2.4 Ineffective Methods

We make a brief mention of the methods that we tried but did not lead to any increase in quality. In particular, for En→De, we built two big R2L models for rescoring ensemble outputs but this technique did not yield any improvement. Therefore, we stopped the experiments in this direction. We also tried to improve the performance of the final ensemble by adding a transformer type language model trained for 2 epochs from the same German News Crawl data we used in other components (116M segments), but this setup did not help in any weight combination we tested either.

In Ja→En, we tested various preprocessing workflows including NFKC Unicode normalization, replacing numbers with placeholders, and also experimented with data selection using only subsets of monolingual data (without News Discussions), subsets of News Commentary selected by topic modelling and n-gram or transformer LM based data selection for tuning, all with no improvement in the results.

4 Results

We submitted one system for each of the five language pairs. In this section we provide evaluation scores for models at important stages in the experiments, which reflect how the models got better

[9] For En→De this meant taking the full OP dataset twice and padding the rest with a subset of OP. This subset was from a language model scored OP data set, see Section 3.2.2 for more details.

[10] We tried 10M and the full 44.7M sets as well.

as we tried various methods for improvement. All results are reported in detokenized BLEU.[11]

4.1 English→German

System	Data	Test sets 2019	2020
M1: Baseline	44.7M	41.9	32.7
M2: M1+BT+CT	64.7M	43.3	34.4
M3: M2+Tbig	232M	44.5	36.9
M4: M3+FT	232M+34.5k	44.8	37.2
M5: M4 ens	232M+34.5k	**46.0**	**37.9**

Table 2: Results for En→De models. The 2020 results are post-submission with the updated (A) reference set.

In Table 2 we present the main stages of the development of the En→De systems. Model 1 was our baseline model and used only the original parallel data[12] (Table 1), which was almost eight times more (already including the full ParaCrawl) than last year, and so the result on the 2019 test set already equaled the performance of our best submission model from last year (Oravecz et al., 2019). Model 2 was the best single base transformer trained from OP extended with 20M tagged back-translated (BT) segments and then with continued training (CT) on the language model scored 20M OP data subset. This yielded substantial improvement but was still far from the best setups. For Model 3, we switched to the big transformer architecture and used the large BT dataset (116M) with the upsampled OP. The training procedure was the same as in the previous system; the first converged model was trained further with the LM-scored OP subset as long as the BLEU score increased. Clearly, this resulted in a more powerful system, further improving the result. The next model (M4) was fine-tuned (FT) with the development set, bringing a small but steady increase. Finally the system we submitted was an ensemble of four M4 models. As last year, a postprocessing step normalizing German punctuation was run on the final hypotheses.

This year the development of the best performing En→De system was dominated by brute force: the more complex and resource demanding architectures performed significantly better, although some careful selection and ranking of the training data also played a role. We managed to train better and better systems as we added more and more resources, and it is very likely that without the limitations in our training environment results could have been further improved.

4.2 Japanese→English

System	Property	Score	Increment
M1	baseline	20.42	–
M2	Bicl. filtering	21.35	+0.93
M3	Unicode filtering	21.53	+0.18
M4	normalization	22.13	+0.60
M5	truecasing	22.07	-0.06
M6	back-translation	23.48	+1.35
M7	balanced BT	23.73	+0.25
M8	fixed big numbers	23.97	+0.24
M9	big transformer	25.39	+1.42
M10	tuned with devset	**27.58**	+2.19

Table 3: Results for Ja→En models. The BLEU score is measured on the development set.

Table 3 summarizes the results of the Ja→En experiments. We trained more than 20 different models from which we present those that produce some increment in the BLEU score. The M1 baseline model was trained from the original parallel data, 13.4 million segments from the 7 constrained resources. This baseline already contained some minimal filtering of duplicates, deletion of markup etc. The M2 model was filtered with Bicleaner (Sánchez-Cartagena et al., 2018), where the filter model was built from this training data. In the M3 system, we used a Unicode range filter, leaving segments containing text using characters only from 35 Unicode character ranges out of the possible 150. In the M4 model, this Unicode filtering was applied before building the Bicleaner filter model. The M5 model used truecasing on the English training and translation data. In M6, synthetic data from back-translation of the monolingual English News Crawl (33M), News Discussion (30M) and News Commentary (0.6M) was added (and tagged). The M7 model contains the same data but the original parallel data was upsampled 3 times to keep a 1:1 ratio to the back-translated data. In M8, we normalized big Japanese numbers to match with

[11] sacreBLEU signatures: `BLEU+case.mixed+lang.en-de+numrefs.1+smooth.exp+tok.13a+version.1.4.12`

[12] We trained only with unique segments, this accounts for the 1.7M decrease from the 46.43M in Table 1.

millions and billions, which were frequently used in the news domain. M9 was a big transformer model built on 4 V100 GPUs. In model M10 (submitted), we tuned the big transformer model on the development set.

4.3 English→Polish

System	Data	Test sets 2020d	2020
M1: Baseline	8.00M	22.2	22.5
M2: M1+BTnews	11.0M	23.3	22.8
M3: M2+BT-Comm-Cr	13.0M	23.4	23.0
Unconstrained			
M4: M3+OPUS+news	53.1M	24.2	23.8
M5: M4+Tbig	53.1M	26.0	24.9
M6: M5 ens	53.1M	26.0	25.0
M7: M6+FT	53.1M	–	**27.2**

Table 4: Results for En→Pl models. The 2020 results are post-submission.

Table 4 presents the main stages of the development of the En→Pl systems. Model 1 was a base Transformer and used only the original parallel data (Table 1). Model 2 included the back-translated News Crawl data, and Model 3 had the addition of the back-translated Common Crawl subset. Each step gave only a very modest improvement. At this stage, we tried to make use of additional data sets and switched to experimenting with unconstrained systems. For Model 4, we added 40M segments of filtered OPUS parallel data, and a small amount of monolingual Polish proprietary data that was back-translated into English. Model 5 is similar to M4 but it is a big transformer, and Model 6 is an ensemble built of three M5 models trained from different seeds. All models for the ensemble were fine-tuned for 24 epochs on 5.5k of domain-specific data consisting of a thousand sentences from the development set plus the manually selected back-translated proprietary news data.

4.4 Russian→English

Table 5 gives a summary of the development stages of the Ru→En systems. M1 and M2 are our baseline systems. Initially, the WikiMatrix data (WM) for Russian was corrupt and we built a baseline without it. After a usable version was provided, we trained another baseline system. M3 included some

System	Data	Test sets 2019	2020
M1: Baseline	6.40M	37.3	33.7
M2: M1+WM	9.80M	38.9	35.3
M3: M2+BT	98.5M	39.1	37.2
M4: M3+Tbig ens	98.5M	40.1	38.0
M5: M3+Tbig+FT1	98.5M	39.6	36.6
M6: M3 ens+FT2	98.5M	–	**37.5**

Table 5: Results for Ru→En models. The 2020 results are post-submission.

50M of back-translated News Crawl and News Discussions data and the OP data of M2 upscaled to a 1:1 ratio to the back-translated data. M4 is an ensemble of 3 big transformer models trained with the same workflow as M3 but with different seeds. M5 is a single big transformer (one of the three in M4) that was fine-tuned for 6 epochs on the 2012–2018 development sets. Finally, M6 is a 3 model ensemble of the fine-tuned models from M4, but for submission fine-tuned on the 2012–2019 development sets.

4.5 English→Czech

System	Parallel data	Test sets 2019	2020
M1: Baseline	45.0M	26.5	31.4
M2: M1+BT	166M	26.8	32.2
M3: M2+Tbig	166M	28.3	33.8
M4: M2+Tbig	166M	28.6	33.7
M5: M3+M4 ens	166M	**28.9**	34.4
M6: sent. seg.	166M	–	**35.7**

Table 6: Results for En→Cs models. The 2020 results are post-submission.

We trained only a few straightforward models for the En→Cs system. The scores in Table 6 give the outcome of the evaluation of 6 simple setups: Model 1 was a base transformer built on the original parallel data (excluding ParaCrawl, which decreased the score). The data for Model 2 was extended with back-translated 2007-2019 News Crawl. In various experiments, the pre 2019 News Crawl data only gave a minor increase in BLEU, the 2019 set was more useful. For the other models, we trained big transformers and built small

ensembles. However, an ensemble of two outperformed the ensemble of three models in the end. We tried continued training on the development sets from the previous years, but it only led to a drop in the score. As a basic post-processing step, we applied double quote and ellipsis normalization. The 2020 test set contained segments with multiple sentences, so in the submission set we performed some sentence segmentation in preprocessing before translation.

4.6 Zero Shot Submissions to the Robustness Task

The best performing En→De (fine-tuned 4 member big transformer ensemble) and Ja→En (fine-tuned big transformer) systems were submitted without any changes as zero shot models for the Robustness Task. Interestingly, these zero shot models (as well as most of the submissions from the other participants), seemed to score better on these very noisy test sets than on the news test sets, suggesting that the training data used was not completely news domain oriented and might already give good support for diverse domains.

5 Conclusion

We described the submissions of the eTranslation team to the WMT 2020 news translation shared task on 5 language pairs: English-German, Japanese-English, English-Polish, Russian-English, and English-Czech. Like last year, we tried to build the best possible systems in a relatively low-resource production environment. But in contrast to last year, we dedicated more resources to certain language pairs, and tried more complex models and utilized significantly more data where possible. In particular, we experimented with various techniques (big transformer models, synthetic data obtained from tagged back-translation, two stage continued training process, ensembling up to 4 models) and obtained significant improvements over baseline models: from 4 to 7 BLEU points depending on the language pair on the 2020 test sets. We ranked competitively in all language pairs, reducing the gap from the best systems significantly from last year.[13] However, the submitted setups cannot be reused in our production environment due to their excessive demands on resources, but lessons learnt from those experiments shall provide valuable insights to improve the eTranslation system under its current constraints.[14]

For the production eTranslation service, with language specific systems for all official EU and EEA languages, finding the right balance between the use of resources in production environments and the best possible performance of models remains a challenge for future work.

[13] For example, in En→De from 3 BLEU points to 0.9.

[14] Current eTranslation resource capacity generally allows only for the baseline models to be trained and deployed, and this, although in a different domain from news, definitely leaves some room for simple "brute force" improvement of the service.

References

Isaac Caswell, Ciprian Chelba, and David Grangier. 2019. Tagged back-translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 53–63, Florence, Italy. Association for Computational Linguistics.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500, Brussels, Belgium. Association for Computational Linguistics.

Vu Cong Duy Hoang, Philipp Koehn, Gholamreza Haffari, and Trevor Cohn. 2018. Iterative back-translation for neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 18–24, Melbourne, Australia. Association for Computational Linguistics.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016. Fasttext.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

Marcin Junczys-Dowmunt. 2019. Microsoft translator at WMT 2019: Towards large-scale document-level neural machine translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 225–233, Florence, Italy. Association for Computational Linguistics.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121. Association for Computational Linguistics.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual*

Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 66–75. Association for Computational Linguistics.

Minh-Thang Luong and Christopher Manning. 2015. Stanford neural machine translation systems for spoken language domains. In *Proceedings of the International Workshop on Spoken Language Translation (IWSLT)*, pages 76–79.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook FAIR's WMT19 news translation task submission. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 314–319, Florence, Italy. Association for Computational Linguistics.

Csaba Oravecz, Katina Bontcheva, Adrien Lardilleux, László Tihanyi, and Andreas Eisele. 2019. eTranslation's submissions to the WMT 2019 news translation task. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 320–326, Florence, Italy. Association for Computational Linguistics.

Víctor M. Sánchez-Cartagena, Marta Bañón, Sergio Ortiz-Rojas, and Gema Ramírez. 2018. Prompsit's submission to WMT 2018 parallel corpus filtering shared task. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 955–962, Belgium, Brussels. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

The ADAPT System Description for the WMT20 News Translation Task

Venkatesh Balavadhani Parthasarathy[‡], Akshai Ramesh[‡], Rejwanul Haque and Andy Way
The ADAPT Centre, [‡]School of Computing
Dublin City University, Dublin, Ireland
venkatesh.balavadhaniparthasa2,akshai.ramesh2@mail.dcu.ie
rejwanul.haque,andy.way@adaptcentre.ie

Abstract

This paper describes the ADAPT Centre's submissions to the WMT20 News translation shared task for English-to-Tamil and Tamil-to-English. We present our machine translation (MT) systems that were built using the state-of-the-art neural MT (NMT) model, Transformer. We applied various strategies in order to improve our baseline MT systems, e.g. monolingual sentence selection for creating synthetic training data, mining monolingual sentences for adapting our MT systems to the task, hyperparameters search for Transformer in low-resource scenarios. Our experiments show that adding the aforementioned techniques to the baseline yields an excellent performance in the English-to-Tamil and Tamil-to-English translation tasks.

1 Introduction

The ADAPT Centre participated in the News Translation Shared Task of the Fifth Conference of Machine Translation (WMT20) in the English-to-Tamil and Tamil-to-English language directions. To build our neural MT systems we used the Transformer model (Vaswani et al., 2017). Our strategies to build the competitive MT systems for the task include applying the state-of-the-art data augmentation approaches (e.g. (Sennrich et al., 2016a; Caswell et al., 2019)), selecting "pseudo" in-domain monolingual sentences for the creation of synthetic bitexts, mining monolingual source and target sentences for the adaptation of neural MT (NMT) systems, finding the optimal set of hyperparameters for Transformer as far as low-resource translation is concerned.

The remainder of the paper is organized as follows. In Section 2, we present our approaches for the MT system building. Section 3 first presents details of the data sets used and then presents the evaluation results with some discussions, while Section 4 concludes our work with avenues for future work.

2 Our Strategies to improve MT Systems

2.1 Data Augmentation

The data augmentation methods (Sennrich et al., 2016a; Zhang and Zong, 2016; Burlot and Yvon, 2018; Poncelas et al., 2018; Bogoychev and Sennrich, 2019; Caswell et al., 2019; Chen et al., 2019), which usually employ the unlabeled monolingual data in addition to limited bitexts, can positively impact the MT system's performance and are very popular among the MT developers and researchers (Barrault et al., 2019). In other words, use of augmented bitexts that include synthetic data to improve a NMT system is nowadays a common practice, especially in the under-resource scenarios. The synthetic training data whose target-side sentences are original is more effective for domain text translation and generation of fluent translations. In this task, in order to improve our baseline Transformer models, we augmented our training data with the target-original synthetic data. As in Caswell et al. (2019), in order to let the NMT model know that the given source is synthetic, we tag the back-translated source sentences with an extra token.

Note that we also tried applying the so-called self-training[1] strategy (Ueffing et al., 2007) to improve our NMT systems. However, this method does not bring any improvements in the Tamil-to-English translation task, and deteriorates the performance of the MT systems in the English-to-Tamil translation task.

Iterative generation and training on synthetic data can yield increasingly better NMT systems,

[1]Synthetic data for training is created by the MT system itself (i.e. source-side is original) (Zhang and Zong, 2016; Burlot and Yvon, 2018).

Proceedings of the 5th Conference on Machine Translation (WMT), pages 262–268
Online, November 19–20, 2020.

especially in low-resource scenarios (Hoang et al., 2018; Chen et al., 2019). Similarly, in order to produce our final English-to-Tamil and Tamil-to-English MT systems, we performed iterative training by back-translating new monolingual data with the updated MT system and appending the resultant synthetic data to the original training data in each iteration.

2.2 Selecting pseudo In-Domain Sentences

In an attempt to improve the quality of our NMT engines, we extracted monolingual sentences from large monolingual data that are similar to the styles of the in-domain data. Sentences of a large monolingual corpus similar to the in-domain sentences when selected based on the perplexity according to an in-domain language model were found to be effective in MT (Gao et al., 2002; Yasuda et al., 2008; Foster et al., 2010; Axelrod et al., 2011; Toral, 2013). As for NMT training, we believe that synthetic parallel data created using pseudo in-domain sentences can be better alternatives than those selected randomly. Accordingly, we select "pseudo" in-domain sentences from a large monolingual corpus based on the perplexity scores according to the in-domain language models. The extracted sentences are then back-translated with a target-to-source MT system to form synthetic training data.

2.3 Mining Monolingual Sentences for the Adaptation of the NMT models

Chinea-Ríos et al. (2017) demonstrated that in case of specialised domains or low-resource scenarios where parallel corpora are scarce sentences of a large monolingual data that are more related to the test set sentences to be translated could be effective for fine-tuning the original general domain NMT model. They select those instances from large monolingual corpus whose vector-space representation is similar to the representation of the test set instances. The selected sentences are then automatically translated by an NMT system that is trained on a general domain data. Finally, the NMT system is fine-tuned with the resultant synthetic data. In a similar line of research, it has also been shown that an NMT system built on general domain data can be fine-tuned using just a few sentences (Farajian et al., 2017, 2018; Wuebker et al., 2018; Huck et al., 2019).

In our case, since English–Tamil is a low-resource language-pair and have a little amount of bitexts pertaining to the targeted domain (News), we followed Chinea-Ríos et al. (2017) and mined those sentences from large monolingual data that can be beneficial for fine-tuning the original NMT models. In addition to mining source-side sentences (Chinea-Ríos et al., 2017), we also mined target language sentences from large monolingual corpus (Huck et al., 2019) when English is the source language. However, our selection methods are different to those of the other papers (Chinea-Ríos et al., 2017; Farajian et al., 2017, 2018; Wuebker et al., 2018; Huck et al., 2019) and are described below.

Terms are usually indicators of the nature of a domain and play a critical role in domain-specific MT (Haque et al., 2020). The target translation could lose its meaning if the terminology translation is not dealt with care. Therefore, we focused on mining those sentences from a large monolingual corpus that contain domain terms. For this, we made use the approach of Rayson and Garside (2000); Haque et al. (2014, 2018) for identifying terms in the test set which is to be translated. This term extraction method performs well even on a small amount of sentences (Haque et al., 2014, 2018). The goal is to identify those words which are most indicative (or characteristic) of the test corpus compared to a reference corpus. Haque et al. (2014, 2018) used a large corpus which is generic in nature as a reference corpus. We adopted their approach and used a large generic corpus in order to identify terms in the test set. Additionally, in our second setup, we used the training set on which the NMT systems were trained as the reference corpus. The intuition is to extract those terms or sequence of words from the test set that do not occur or rarely occur in the training set and convey representativeness of the test set. We merged the two sets of terms extracted following the two setups above. Given the resultant list of terms, we mine sentences from monolingual corpus.

We observed that the WMT20 News development text contains many named entities (NEs) and many of them are out-of-vocabulary items. We also found that our initial MT systems miserably failed to translate many NEs. Therefore, we used Stanford named entity recogniser (NER)[2] (Finkel et al., 2005) in order to identify NEs in the English test set. As above, we used the extracted NEs in order to mine sentences from a large monolingual corpus.

[2]https://nlp.stanford.edu/software/CRF-NER.html

We build an English-to-Tamil transliteration system and the extracted English NEs were transliterated into Tamil. Note that we took 5-best Tamil translations for an English NE as in Huck et al. (2019). These Tamil NEs were then used to mine Tamil sentences from a large target monolingual corpus.

In order to build the English-to-Tamil transliteration system, we used the 2016 Named Entity Transliteration Shared Task (NEWS) dataset[3] (Duan et al., 2016). We used our in-house machine transliteration tool (Haque et al., 2009) in order to prepare the English-to-Tamil transliteration system.

We could not apply this strategy in the Tamil-to-English translation task since there is no publicly available NER for Tamil. The source and target sentences that have been mined are translated with the final source-to-target and target-to-source NMT systems, respectively. This results in a set of synthetic sentence-pairs. Source sentences whose target-side is original are tagged with a special token (Caswell et al., 2019) (cf. Section 2.1). As in Chinea-Ríos et al. (2017), the original MT system is finally fine-tuned on these synthetic segment-pairs.

For mining monolingual sentences we create an efficient Trie structure given the large monolingual data. The idea is to store indices of the sentences (i.e. we restrict this number to 50) for each n-gram (upto trigram) of the corpus. Given the domain terms of the in-domain text, we can instantly retrieve the sentences from corpus.

2.4 Tuning Hyperparameters for Transformer

The NMT systems are Transformer models (Vaswani et al., 2017). To build our NMT systems, we used the MarianNMT (Junczys-Dowmunt et al., 2018) toolkit. The tokens of the training, evaluation and validation sets are segmented into sub-word units using Byte-Pair Encoding (BPE) (Sennrich et al., 2016b). Since English and Tamil are written in their own scripts and have no overlapping characters, BPE is applied individually on the source and target languages. Recently, Sennrich and Zhang (2019) demonstrated that commonly used hyper-parameter configuration do not lead to the best results in low-resource settings. Accordingly, we carried out a series of experiments in order to find the best hyperparameter configuration for Transformer in our low-resource setting.[4] In particular, we played with some of the hyperparameters, and found that the following configuration lead to the best results in our low-resource translation settings: (i) the BPE vocabulary size: 8,000, (ii) the sizes of the encoder and decoder layers: 4 and 6, respectively, (iii) learning-rate: 0.0005, (iv) dropout (Gal and Ghahramani, 2016) between layers: 0.1. As for the remaining hyperparameters, we followed the recommended best setup from Vaswani et al. (2017). The models are trained with the Adam optimizer (Kingma and Ba, 2014), reshuffling the training corpora for each epoch. The early stopping criteria is based on cross-entropy; however, the final NMT system is selected as per the highest BLEU score on the validation set. The beam size for search is set to 12. We make our final NMT model with ensembles of 8 models that are sampled from the training run.

3 Experiments and Results

3.1 Data sets

This section presents the data sets that were used for system building. We used the monolingual and bilingual data provided by the WMT20 task organisers only. No external data has been used for the MT system building. Table 1 presents the corpus statistics. The parallel corpora released by

	Parallel Data		
	sentences	words (EN)	words (TA)
train	350,142	6,489,872	5,763,047
test	1,000	23,259	17,966
dev.	989	23,415	17,901
	Monolingual Data		
English	17M	Tamil	31M

Table 1: The data statistics.

WMT20 for the English–Tamil task are from different sources (e.g. Tanzil v1[5] (Tiedemann, 2012), WikiMatrix[6] (Schwenk et al., 2019) and PMIndia[7] (Haddow and Kirefu, 2020)). We merged segment-pairs of all data sources, and after applying standard cleaning scripts to the data we are left with

[3] http://workshop.colips.org/news2016

[4] This set of experiments were conducted on English-to-Tamil only and using the bitexts only. The best hyperparameter setup found in this task is used in the reverse translation task.

[5] http://opus.nlpl.eu/Tanzil-v1.php

[6] https://ai.facebook.com/blog/wikimatrix/

[7] http://data.statmt.org/pmindia

350K parallel segments (cf. row 3 of Table 1). As for the monolingual data, we used News-Crawl[8] and CommonCrawl[9] corpora (cf. last row of Table 1).

We observed that the corpora of one language (say, Tamil) contains sentences of other languages (e.g. English), so we use a language identifier[10] in order to remove such noise. In order to perform tokenisation for English and Tamil texts, we used the standard tool[11] in the Moses toolkit.

WMT20 released a development set of 1,989 sentences (*newsdev2020*) whose domain is naturally news. We used 1,000 sentences from *newsdev2020* as test set, and we call the test set *newstest1k*. The remaining sentences (989) are treated as the validation set.

3.2 The Baseline MT Systems

The BLUE scores of the NMT systems trained on the authentic parallel corpus (cf. Table 1) are reported in Table 2. These BLEU scores represent the MT systems that were trained following the best hyperparameter settings described in Section 2.4. Note that these MT systems serve our baselines. We refer the baseline MT system as Base. We see from Table 2 that the English-to-Tamil and

	newstest1k
English-to-Tamil	5.81
Tamil-to-English	12.20

Table 2: The BLEU score of the baseline MT systems.

Tamil-to-English MT systems produce 5.81 and 12.20 BLEU scores, respectively, on the respective test sets. As expected, the translation quality from the morphologically-rich to morphologically-poor language improves.

3.3 The Improved MT Systems

We applied the pseudo in-domain sentence selection strategy described in Section 2.2 to the monolingual corpora (cf. Table 1), and considered the top-scored sentences for back-translation. Note that the in-domain language models for sentence selection were built on the PMI data whose domain is news in nature. The development set sentences (998 sentences; cf. Table 2) were also appended to the PMI data. The BLEU scores on the MT systems trained on the augmented training data are presented in Table 3. We stopped the iterative training process (cf. Section 2.1) when there were no significant improvements in terms of the test set BLEU scores. This training process provides us with the improved MT systems. As can be seen from Table 3, the final MT systems surpass the respective baseline MT systems with large margins.

	newstest1k BLEU	newstest2020 SacreBLEU
English-to-Tamil		
Base+ 800K	8.12	
Base+ 1.5M	9.35	
Base+ 2.7M	9.45	5.4
Tamil-to-English		
Base + 800K	18.33	
Base + 1.5M	18.52	
Base + 2.7M	19.41	
Base + 3.3M	19.91	14.7

Table 3: The BLEU scores of the MT systems trained on the interactively augmented training data.

We translate the blind test sets (newstest2020) for the English-to-Tamil and Tamil-to-English translation tasks released by WMT20 by the best MT systems (cf. Table 3). The blind test sets for the English-to-Tamil and Tamil-to-English tasks contain 6,988 and 997 segments, respectively. The sacreBLEU (Post, 2018) scores of the best NMT systems on newstest2020 are shown in the last column of Table 3.[12]

3.4 Fine-tuning the best NMT systems

This section presents the MT systems that were prepared by the adaptation technique described in Section 2.3. We mine the source and target monolingual sentences from the large monolingual corpora given the terms and NEs (and transliterated NEs) extracted from newstest1k.[13] As described in Section 2.3, synthetic data is created by translating the source and target sentences by the target-to-source and source-to-target MT systems (cf. Table 3; the best MT systems), respectively. Finally, the

[8] http://data.statmt.org/news-crawl

[9] http://data.statmt.org/news-discussions/

[10] cld2: https://github.com/CLD2Owners/cld2

[11] https://github.com/moses-smt/mosesdecoder/blob/master/scripts/tokenizer/tokenizer.perl

[12] The SacreBLEU scores were taken from OCELoT https://ocelot.mteval.org

[13] Note that NEs were extracted from the English text only.

best MT system is fine-tuned on the synthetic data. The BLEU scores of the adapted MT systems on newstest1k are reported in Section 4. When we compare the original MT systems reported in Table 3 with the adapted MT systems, we see that (i) the English-to-Tamil adapted MT system produces a 1.55 BLEU points (corresponding to 16.4% relative) improvement over the the original English-to-Tamil MT system, and (ii) the Tamil-to-English adapted MT system produces a 1.41 BLEU points (corresponding to 7.08% relative) improvement over the the original Tamil-to-English MT system. The improvements are statistically significant.

	newstest1k BLEU	newstest2020 SacreBLEU
English-to-Tamil	10.80	6.1
Tamil-to-English	21.32	15.8

Table 4: The BLEU and SacreBLEU scores of the adapted MT systems on newstest1k and newstest2020, respectively.

As above, we create the English-to-Tamil and Tamil-to-English adapted MT systems for the blind test sets. Then, we translate the blind test sets with the adapted MT systems. The sacreBLEU (Post, 2018) scores of the adapted MT systems on newstest2020 are shown in the last column of Table 4. Again, the adaption strategy brings about moderate improvements over the original MT systems, i.e. a 0.7 SacreBLEU points (corresponding to 13% relative) improvement for the English-to-Tamil translation and a 1.1 SacreBLEU points (corresponding to 7.5% relative) improvement for the Tamil-to-English translation.

4 Conclusion

This paper presents the ADAPT system description for the WMT20 News Translation Shared Task. We participated in the English-to-Tamil and Tamil-to-English tasks. English–Tamil is a low-resource language-pair and we used the data provided by the WMT20 organisers only. Given the limited resources provided for the tasks, we aimed to build the competitive translation systems for the competition. For this, we applied a variety of strategies, e.g. iterative data augmentation, selection of pseudo in-domain sentences, and a novel strategy for the adaptation of the NMT models to the task. We found that the systematic addition of these techniques to baseline yields excellent improvements over the baseline.

This paper presented an effective adaptation method for the NMT systems. This method is found to be effective as far as the translation task we participated in is concerned. In the future, we aim to test on-the-fly adaptation method (Farajian et al., 2017, 2018) to translate domain texts.

Acknowledgments

The ADAPT Centre for Digital Content Technology is funded under the Science Foundation Ireland (SFI) Research Centres Programme (Grant No. 13/RC/2106) and is co-funded under the European Regional Development Fund. This project has partially received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 713567, and the publication has emanated from research supported in part by a research grant from SFI under Grant Number 13/RC/2077.

References

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. 2011. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 355–362, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Loïc Barrault, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. 2019. Findings of the 2019 conference on machine translation (wmt19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 1–61, Florence, Italy. Association for Computational Linguistics.

Nikolay Bogoychev and Rico Sennrich. 2019. Domain, translationese and noise in synthetic data for neural machine translation. *arXiv preprint arXiv:1911.03362*.

Franck Burlot and François Yvon. 2018. Using monolingual data in neural machine translation: a systematic study. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 144–155, Belgium, Brussels. Association for Computational Linguistics.

Isaac Caswell, Ciprian Chelba, and David Grangier. 2019. Tagged back-translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 53–63, Florence, Italy. Association for Computational Linguistics.

Peng-Jen Chen, Jiajun Shen, Matthew Le, Vishrav Chaudhary, Ahmed El-Kishky, Guillaume Wenzek, Myle Ott, and Marc'Aurelio Ranzato. 2019. Facebook AI's WAT19 Myanmar-English translation task submission. In *Proceedings of the 6th Workshop on Asian Translation*, pages 112–122, Hong Kong, China.

Mara Chinea-Ríos, Álvaro Peris, and Francisco Casacuberta. 2017. Adapting neural machine translation with parallel synthetic data. In *Proceedings of the Second Conference on Machine Translation*, pages 138–147, Copenhagen, Denmark. Association for Computational Linguistics.

Xiangyu Duan, Rafael Banchs, Min Zhang, Haizhou Li, and A. Kumaran. 2016. Report of NEWS 2016 machine transliteration shared task. In *Proceedings of the Sixth Named Entity Workshop*, pages 58–72, Berlin, Germany. Association for Computational Linguistics.

M Amin Farajian, Nicola Bertoldi, Matteo Negri, Marco Turchi, and Marcello Federico. 2018. Evaluation of terminology translation in instance-based neural mt adaptation. In *Proceedings of the European Association for Machine Translation*.

M Amin Farajian, Marco Turchi, Matteo Negri, and Marcello Federico. 2017. Multi-domain neural machine translation through unsupervised adaptation. In *Proceedings of the Second Conference on Machine Translation*, pages 127–137.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by Gibbs sampling. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 363–370, Ann Arbor, Michigan. Association for Computational Linguistics.

George Foster, Cyril Goutte, and Roland Kuhn. 2010. Discriminative instance weighting for domain adaptation in statistical machine translation. In *Proceedings of the 2010 conference on empirical methods in natural language processing*, pages 451–459.

Yarin Gal and Zoubin Ghahramani. 2016. A theoretically grounded application of dropout in recurrent neural networks. *CoRR*, abs/1512.05287.

Jianfeng Gao, Joshua Goodman, Mingjing Li, and Kai-Fu Lee. 2002. Toward a unified approach to statistical language modeling for chinese. *ACM Transactions on Asian Language Information Processing (TALIP)*, 1(1):3–33.

Barry Haddow and Faheem Kirefu. 2020. Pmindia–a collection of parallel corpora of languages of india. *arXiv preprint arXiv:2001.09907*.

Rejwanul Haque, Sandipan Dandapat, Ankit Kumar Srivastava, Sudip Kumar Naskar, and Andy Way. 2009. English-Hindi transliteration using context-informed PB-SMT: the DCU system for NEWS 2009. In *Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration (NEWS 2009)*, pages 104–107, Suntec, Singapore. Association for Computational Linguistics.

Rejwanul Haque, Mohammed Hasanuzzaman, and Andy Way. 2020. Analysing terminology translation errors in statistical and neural machine translation. *Machine Translation (in press)*, 34.

Rejwanul Haque, Sergio Penkale, and Andy Way. 2014. Bilingual termbank creation via log-likelihood comparison and phrase-based statistical machine translation. In *Proceedings of the 4th International Workshop on Computational Terminology (Computerm)*, pages 42–51, Dublin, Ireland.

Rejwanul Haque, Sergio Penkale, and Andy Way. 2018. TermFinder: log-likelihood comparison and phrase-based statistical machine translation models for bilingual terminology extraction. *Language Resources and Evaluation*, 52(2):365–400.

Vu Cong Duy Hoang, Philipp Koehn, Gholamreza Haffari, and Trevor Cohn. 2018. Iterative back-translation for neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 18–24, Melbourne, Australia. Association for Computational Linguistics.

Matthias Huck, Viktor Hangya, and Alexander Fraser. 2019. Better oov translation with bilingual terminology mining. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5809–5815.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Alberto Poncelas, Dimitar Shterionov, Andy Way, Gideon Maillette de Buy Wenniger, and Peyman Passban. 2018. Investigating backtranslation in neural machine translation. In *Proceedings of The 21st Annual Conference of the European Association for Machine Translation (EAMT 2018)*, pages 249–258, Alicante, Spain.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

Paul Rayson and Roger Garside. 2000. Comparing corpora using frequency profiling. In *The workshop on comparing corpora*, pages 1–6.

Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán. 2019. Wikimatrix: Mining 135m parallel sentences in 1620 language pairs from wikipedia. *arXiv preprint arXiv:1907.05791*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany.

Rico Sennrich and Biao Zhang. 2019. Revisiting low-resource neural machine translation: A case study. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 211–221, Florence, Italy. Association for Computational Linguistics.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC'2012)*, pages 2214–2218, Istanbul, Turkey.

Antonio Toral. 2013. Hybrid selection of language model training data using linguistic information and perplexity. In *Proceedings of the second workshop on hybrid approaches to translation*, pages 8–12.

Nicola Ueffing, Gholamreza Haffari, and Anoop Sarkar. 2007. Transductive learning for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 25–32, Prague, Czech Republic. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 6000–6010.

Joern Wuebker, Patrick Simianer, and John DeNero. 2018. Compact personalized models for neural machine translation. *arXiv preprint arXiv:1811.01990*.

Keiji Yasuda, Ruiqiang Zhang, Hirofumi Yamamoto, and Eiichiro Sumita. 2008. Method of selecting training data to build a compact and efficient translation model. In *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-II*.

Jiajun Zhang and Chengqing Zong. 2016. Exploiting source-side monolingual data in neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545, Austin, Texas. Association for Computational Linguistics.

CUNI English-Czech and English-Polish Systems in WMT20: Robust Document-Level Training

Martin Popel
Charles University, Faculty of Mathematics and Physics,
Institute of Formal and Applied Linguistics,
Malostranské náměstí 25, 118 00 Prague, Czech Republic
popel@ufal.mff.cuni.cz

Abstract

We describe our two NMT systems submitted to the WMT 2020 shared task in English↔Czech and English↔Polish news translation. One system is sentence level, translating each sentence independently. The second system is document level, translating multiple sentences, trained on multi-sentence sequences up to 3000 characters long.

1 Introduction

In this paper, we describe our two NMT systems submitted to the WMT 2020 news translation shared task: "CUNI-Transformer" (Charles University Transformer, sentence-level) and "CUNI-DocTransformer" (document-level). We trained them for English↔Czech and the former one also for English↔Polish (no parallel document-level data was provided for English-Polish, thus we could not train the latter one).

2 Common settings

Both our systems are implemented in the Tensor2Tensor framework (Vaswani et al., 2018) and have the same Transformer (Vaswani et al., 2017) architecture – transformer_big with 12 encoder layers instead of the default 6 (while keeping 6 layers in the decoder). The 32k joint English-Czech subword vocabulary is exactly the same as used by Popel (2018) and Popel et al. (2019), which are the systems we submitted to WMT in the last two years. Also most of the hyperparameters (except for the encoder depth) and the training regime are the same.

The main improvement of our sentence-level system relative to our last-year submission stems from using slightly larger and better-filtered training data – CzEng 2.0 (Kocmi et al., 2020b) with 61M authentic parallel and 127M synthetic (back-translated) sentences (see Table 1), instead of CzEng 1.7 with 57M authentic parallel sentences.

data set	sentence pairs (M)	words (M) EN	CS
authentic	61	617	702
EN-mono (NewsCrawl 2016–2018)	76	1296	1474
CS-mono (NewsCrawl 2013–2018)	51	700	833
total	188	2613	3009

Table 1: Training data sizes (in millions). All the data are taken from CzEng 2.0.

We also enlarged our development-test set: we concatenated WMT newstest 2008–2018, instead of using newstest2016 only. WMT news tests before 2020 did not have paragraph boundaries marked. We thus prepared a version of our dev-set where we joined together several consecutive sentences randomly (except for titles not ended by a punctuation) to simulate WMT2020 paragraph-level setting.

Our document-level system was further improved as described in Sections 3 and 4.

3 Document-level training

Our last-year document-level submission (Popel et al., 2019) introduced a method of training-data context augmentation, where multiple consecutive sentences (within original documents) are merged together into multi-sentence sequences (of parallel source-target data). The sentences within each sequence are separated with a special token, so that we can easily extract the sentence alignment after decoding. The length of the sequences was limited by 1000 characters and 200 subwords (i.e. any sequences longer than any of the limits in either source or target were discarded from training).

An important aspect of this method is that it extracts all possible sequences from the document-level training data. For example, given a document

Proceedings of the 5th Conference on Machine Translation (WMT), pages 269–273
Online, November 19–20, 2020.

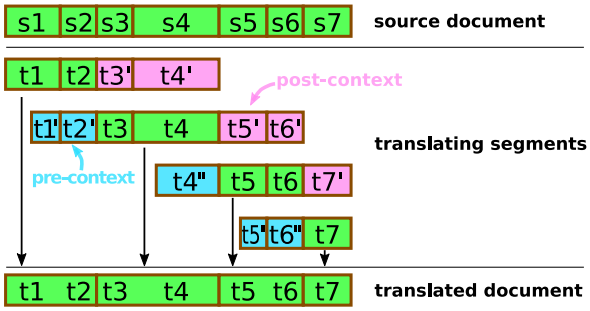


Figure 1: Decoding overlapping multi-sentence sequences with our document-level model. Note that the pre-context part may start not only on sentence boundaries (it improves the results slightly according our initial experiments).

with 5 sentences s1–s5, we extract sequences s1, s1–2, s1–s3, s2, s2–s3, s2–s4, s3, s3–s4, s3–s5, s4, s4–s5 and s5, while ignoring sequences s1–s4, s1–s5 and s2–s5 because these are longer than the limits. Note that this way of context augmentation implicitly upsamples sentences from longer documents relative to sentences from shorter documents. A sentence appearing in N windows of at most 1000 characters is present N times in the augmented training data.

Thus this year, we simply sample non-overlapping sequences of sentences: s1–s3, s4–s5. There are many documents shorter than the limits in CzEng 2.0, including many single-sentence documents (from sources without document-level annotation). Thus, there are naturally occurring trainng sequences which are shorter than the limits and we checked the model is capable of translating also single sentences.

In addition to this change in data sampling, we increased the sequence length limit to 3000 characters and 750 subwords.

4 Document-level decoding

There are many possible ways how to use document-level models (trained as described in the previous section) at decode time.

- We can translate single sentences, thus not using the advantage of document-level training. This may serve as a baseline for comparison with document-level decoding and we used this for our last-year sentence-level submission "Transformer T2T 2019" (Popel et al., 2019).

- We can split each input document into non-overlapping multi-sentence sequences.

- We can split each input document into overlapping multi-sentence sequences (with so-called pre-context and post-context parts, which are ignored in the final translation) as suggested by Popel et al. (2019) and explained in Figure 1.

- We can use the overlapping sequences for some kind of consensus decoding or ensembling. We have not tried this option yet.

Because of the increased limits of training sequences, we increased also the decoding limits two times: pre-context of up to 400 characters, main content of up to 1000 characters and post-context of up to 1800 characters minus the length of the pre-context and main content.

5 Robust training with noising

To make the model more robust to real-world user-generated data, we added a noise to the training data. We followed an approach of Náplava and Straka (2019) and made the source side of the training data more noisy by introducing both grammatical and spelling errors. The basic set of noising operations introducing grammatical errors consisted of the following operations: token replacement with one of its spelling dictionary proposals, token deletion and insertion and swapping of two nearby words. Moreover, we also allowed to replace phrases with one of their most frequent variants, add or delete punctuation and allowed to strip diacritics.

We applied this technique only to our Czech→English sentence-level system by noising the source=Czech side of both the authentic and synthetic parallel data. In preliminary experiments, we observed substantial improvements on artificially noised dev sets, but slight worsenings on WMT dev sets, which contain just a very small amount of typos and other errors (on the source side). We thus decided to mix the noised training data with the original unnoised data 1:1. This resulted in approximately the same BLEU on the original dev sets as without noising, while keeping the improved results on artificially noised dev sets.

For time constraints, we decided to not use any noising in the document-level Czech→English training, as well as in our English→Czech and English↔Polish systems.

6 Results

In Tables 2–5, we report BLEU scores on the newstest2020 for all the systems submitted to WMT.[1] For English→Czech and English→Polish (Tables 2 and 4), we report also gender coreference accuracy scores based on WinoMT testset results (Kocmi et al., 2020a). For English→Czech, we report also manual document-level quality evaluation by Zouhar et al. (2020) of 269 WMT2020 test-suites sentences (i.e. not sentences from newstest2020). The official manual evaluation on newstest2020 is not available yet.

We can see that while our DocTransformer was not the best system according to BLEU, it scored well according to the other two reported metrics, being the best system in English→Czech and the second-best in English→Polish. This could be caused by the low reliability of BLEU (and other metrics based on similarity with reference) for high-quality MT, or by the domain mismatch – the test-suites contain also other domains than news. Unfortunatelly, we cannot answer this question before further analysis using the official WMT manual evaluation, once it is done and published.

Finally, we present several translation examples in the Appendix. The source English documents were taken from the WMT2019 newstest and the same examples were selected already by Popel et al. (2019). Table 6 shows three examples where the document-level model corrects a lexical error of our 2019 sentence-level model. Interestingly, two of these errors were fixed also by our this-year sentence-level model, showing that the cross-sentence context is not *necessary* for correct translation of these examples. Table 7 shows an error of our 2019 document-level model, which is not present in this-year models.

7 Conclusion

We suceeded to improve our baseline system CUNI-T2T-2018 (Popel et al., 2019) by using better training data, doubling the encoder depth (to 12 layers) and by robust training with source-side noising. While all these three techniques are well-known, we show improvements in improving the last-year WMT state of the art in English-Czech translation. We improved also our document-level system (CUNI-DocTransformer) by more careful data sam-

[1]The SacreBLEU signature is BLEU+case.mixed+lang.$src-$trg+numrefs.1+smooth.exp+test.wmt20+tok.13a+version.1.4.13.

system	BLEU cased	g. coref accuracy	TS fluency × adequacy
Online-B	**41.11**	(11) 56.9	(4) 83.3
OPPO	36.78	(3) 78.7	(2) 84.2
SRPOL	36.46	(2) 81.2	(5) 82.2
UEDIN-CUNI	36.27	(6) 72.5	(8) 79.5
CUNI-DocTransformer	35.67	**(1) 83.6**	**(1) 85.1**
eTranslation	35.67	(8) 70.9	(7) 80.5
CUNI-Transformer	35.40	(4) 78.0	(3) 83.4
CUNI-T2T-2018	35.08	(5) 77.6	(6) 81.0
Online-A	30.84	(9) 63.3	(9) 78.9
Online-Z	27.96	(7) 72.2	(10) 72.8
Online-G	25.28	(10) 62.0	(11) 71.7
zlabs-nlp	20.25	(12) 49.9	(12) 64.5

Table 2: Evaluation of English→Czech WMT20 systems. The systems are ordered by BLEU, ordering by the other metrics is provided in parentheses. The gender coreference accuracy scores are based on the WinoMT testset results (Kocmi et al., 2020a). The "TS fluency × adequacy" score is based on manual document-level quality evaluation (Zouhar et al., 2020).

system	BLEU cased
OPPO	**29.91**
CUNI-DocTransformer	29.22
Online-B	28.66
CUNI-Transformer	28.55
SRPOL	28.51
UEDIN-CUNI	27.66
Online-A	26.84
CUNI-T2T-2018	26.08
PROMT_NMT	25.57
Online-G	23.91
Online-Z	23.25
zlabs-nlp	21.76

Table 3: Evaluation of Czech→English WMT20 systems.

system	BLEU cased	g. coref accuracy
SRPOL	**27.56**	**(1) 71.2**
eTranslation	27.20	(3) 68.8
Huoshan_Translate	26.09	(8) 65.7
OPPO	25.49	(4–5) 68.2
SJTU-NICT	25.45	(4–5) 68.2
Online-B	25.17	(12) 57.7
Tilde (1430)	24.93	(9) 64.8
NICT_Kyoto	24.91	(10) 64.2
Tilde (1425)	24.87	(11) 63.3
CUNI-Transformer	24.76	(2) 69.8
Online-G	23.73	(6) 67.3
Online-A	23.71	(13) 53.7
Online-Z	20.75	(7) 65.9
zlabs-nlp	18.64	(14) 46.1

Table 4: Evaluation of English→Polish WMT20 systems.

system	BLEU cased
NICT-Rui	**34.55**
Huoshan_Translate	34.44
SRPOL	34.26
Online-B	33.92
OPPO	32.46
SJTU-NICT	32.16
CUNI-Transformer	31.90
NICT_Kyoto	31.85
Online-A	31.79
PROMT_NMT	31.19
Tilde	30.20
Online-G	29.86
Online-Z	28.64
zlabs-nlp	27.77

Table 5: Evaluation of Polish→English WMT20 systems.

pling which is not biased towards sentences from longer documents.

Acknowledgments

This work has been supported by the grant GX20-16819X (LUSyD) of the Grant Agency of the Czech Republic and by the LINDAT/CLARIN project of the Ministry of Education, Youth and Sports of the Czech Republic (LM2015071). The work has been using language resources developed and distributed by the LINDAT/CLARIAH-CZ project of the Ministry of Education, Youth and Sports of the Czech Republic (project LM2018101).

References

Tom Kocmi, Tomasz Limisiewicz, and Gabriel Stanovsky. 2020a. Gender Coreference and Bias Evaluation at WMT 2020. Submitted to WMT2020.

Tom Kocmi, Martin Popel, and Ondrej Bojar. 2020b. Announcing CzEng 2.0 Parallel Corpus with over 2 Gigawords. *arXiv preprint arXiv:2007.03006*.

Jakub Náplava and Milan Straka. 2019. Grammatical error correction in low-resource scenarios. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 346–356.

Martin Popel. 2018. CUNI Transformer Neural MT System for WMT18. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 486–491, Belgium, Brussels. Association for Computational Linguistics.

Martin Popel, Dominik Macháček, Michal Auersperger, Ondřej Bojar, and Pavel Pecina. 2019. English-Czech systems in WMT19: Document-level transformer. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 342–348, Florence, Italy. Association for Computational Linguistics.

Ashish Vaswani, Samy Bengio, Eugene Brevdo, Francois Chollet, Aidan N. Gomez, Stephan Gouws, Llion Jones, Łukasz Kaiser, Nal Kalchbrenner, Niki Parmar, Ryan Sepassi, Noam Shazeer, and Jakob Uszkoreit. 2018. Tensor2tensor for neural machine translation. *CoRR*, abs/1803.07416.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, pages 6000–6010. Curran Associates, Inc.

Vilem Zouhar, Tereza Vojtechova, and Ondrej Bojar. 2020. WMT20 Document-Level Markable Error Exploration. Submitted to WMT2020.

8 Appendix

source	[...] to meet Craig Halkett's header across goal. The hosts were content to let Rangers play in front of them, knowing they could trouble the visitors at set pieces. And that was the manner in which the crucial **goal** came. Rangers conceded a free-kick [...]
T2T-2019-sent	A to byl způsob, jakým přišel rozhodující **cíl** (*aim*).
others	A to byl způsob, jakým přišel rozhodující **gól** (*goal*).
source	Elizabeth Warren Will Take "Hard Look" At Running For President in 2020, Massachusetts Senator Says Massachusetts Senator Elizabeth Warren said on Saturday she would take a "hard look" at running for president following the midterm elections. During a town hall in Holyoke, Massachusetts, Warren confirmed she'd consider **running**. "It's time for women to go to Washington and fix our broken government and that includes a woman at the top," she said, according to The Hill. [...]
T2T-2019-sent	Na radnici v Holyoke v Massachusetts Warrenová potvrdila, že uvažuje o **útěku** (*escape*).
T2T-2019-doc	Na radnici v Holyoke ve státě Massachusetts Warrenová potvrdila, že o **kandidatuře** (*candidacy*) uvažuje.
T2T-2020-sent	Na radnici v Holyoke v Massachusetts Warrenová potvrdila, že zváží **kandidaturu** (*candidacy*).
T2T-2020-doc	Během jednání na radnici v Holyoke ve státě Massachusetts Warrenová potvrdila, že o **kandidatuře** (*candidacy*) bude uvažovat.
source	At 6am, just as Gegard Mousasi and Rory MacDonald were preparing to face each other, viewers in the UK were left stunned when the coverage changed to Peppa Pig. Some were unimpressed after they had stayed awake until the early hours especially for the **fight**. [...]
T2T-2019-sent	Na některé to neudělalo žádný dojem, když zůstali vzhůru až do časných ranních hodin, zvláště kvůli **rvačce** (*brawl*).
T2T-2019-doc	Na některé to neudělalo žádný dojem, když zůstali vzhůru až do ranních hodin, zejména kvůli **zápasu** (*match*).
T2T-2019-sent	Na některé to neudělalo žádný dojem poté, co zůstali vzhůru až do časných ranních hodin, zejména kvůli **boji** (*combat*).
T2T-2020-doc	Někteří nebyli ohromeni poté, co zůstali vzhůru až do ranních hodin, zejména kvůli **zápasu** (*match*).

Table 6: Three examples of errors by T2T-2019-sent (2019 sentence-level model) corrected by the document-level models and in the first two examples also by T2T-2020-sent (sentence-level CUNI-Transformer from this paper).

source	New cancer vaccine can teach the immune system to 'see' rogue cells New cancer vaccine can teach the immune system to 'see' rogue cells and kill them Vaccine teaches immune system to recognise rogue cells as part of treatment Method involves extracting immune cells from a **patient**, altering them in lab They can then 'see' a protein common to many cancers and then reinjected A trial vaccine is showing promising results in **patients** with a range of cancers. One woman treated with the vaccine, which teaches the immune system to recognise rogue cells, saw her ovarian cancer disappear for more than 18 months. The method involves extracting immune cells from a **patient**, altering them in the laboratory so they can "see" a protein common to many cancers called HER2, and then reinjecting the cells.
T2T-2019-sent	[...] buněk z **pacienta** [...] výsledky u **pacientů** [...] buněk z **pacienta** [...]
T2T-2019-doc	[...] buněk z **pacienta** [...] výsledky u **pacientů** [...] buněk od **pacientky** (*female patient*) [...]
T2T-2020-sent	Nová protinádorová vakcína může naučit imunitní systém „vidět" nepoctivé buňky Nová vakcína proti rakovině může naučit imunitní systém „vidět" buňky darebáků a zabít je Vakcína učí imunitní systém rozpoznat nepoctivé buňky jako součást léčby Metoda zahrnuje extrakci imunitních buněk z **pacienta**, jejich úpravu v laboratoři Mohou pak „vidět" bílkovinu, která je společná mnoha druhům rakoviny a pak ji znovu nasadit Zkušební vakcína vykazuje slibné výsledky u **pacientů** s řadou nádorových onemocnění. Jedna žena léčená vakcínou, která učí imunitní systém rozpoznávat nepoctivé buňky, se postarala o to, že jí na více než 18 měsíců zmizela rakovina vaječníků. Metoda spočívá v extrakci imunitních buněk z **pacienta**, jejich modifikaci v laboratoři, aby mohli „vidět" bílkovinu společnou mnoha druhům rakoviny zvanou HER2, a následném reinjekci buněk.
T2T-2020-doc	Nová protinádorová vakcína může naučit imunitní systém „vidět" darebácké buňky Nová protinádorová vakcína může naučit imunitní systém „vidět" darebácké buňky a zabít je Vakcína učí imunitní systém rozpoznávat darebácké buňky jako součást léčby Metoda spočívá v odebrání imunitních buněk z **pacienta**, jejich pozměnění v laboratoři Mohou pak „vidět" bílkovinu společnou pro mnoho druhů rakoviny a poté znovu použít Zkušební vakcína vykazuje slibné výsledky u **pacientů** s řadou druhů rakoviny. Jedna žena léčená touto vakcínou, která učí imunitní systém rozpoznávat darebácké buňky, viděla, jak její rakovina vaječníků zmizela na více než 18 měsíců. Metoda spočívá v odebrání imunitních buněk z **pacienta**, jejich pozměnění v laboratoři tak, aby mohly „vidět" bílkovinu společnou pro mnoho druhů rakoviny nazývanou HER2, a poté znovu použít buňky.

Table 7: Example of an inconsistency error by T2T-2019-doc. The other three models are consistent.

Machine Translation for English–Inuktitut with Segmentation, Data Acquisition and Pre-Training

Christian Roest† **Lukas Edman**‡ **Gosse Minnema**‡
Kevin Kelly† **Jennifer Spenader**† **Antonio Toral**‡
†Institute for Artificial Intelligence ‡Center for Language and Cognition,
University of Groningen
The Netherlands

c.roest@student.rug.nl, j.l.edman@rug.nl, g.f.minnema@rug.nl

kevin.kelly@live.se, j.spenader@ai.rug.nl, a.toral.ruiz@rug.nl

Abstract

Translating to and from low-resource polysynthetic languages present numerous challenges for NMT. We present the results of our systems for the English–Inuktitut language pair for the WMT 2020 translation tasks. We investigated the importance of correct morphological segmentation, whether or not adding data from a related language (Greenlandic) helps, and whether using contextual word embeddings improves translation. While each method showed some promise, the results are mixed.

1 Introduction

This paper presents the neural machine translation (NMT) systems submitted by the University of Groningen to the WMT 2020 translation task[1] between Inuktitut and English in both directions (EN↔IU), describing both constrained and unconstrained systems where we investigated the following research questions:

- RQ1. Does morphological segmentation benefit translation with polysynthetic languages? Existing NMT research showed that morphological segmentation outperforms byte-pair encoding (BPE) (Sennrich et al., 2016) for some agglutinative languages. For example, rule-based morphological segmentation improved English-to-Finnish translation (Sánchez-Cartagena and Toral, 2016). and unsupervised morphological segmentation improved Turkish-to-English translation (Ataman et al., 2017). We investigate if morphological segmentation also improves translation performance for polysynthetic languages, and if effects differ depending on translation direction.
- RQ2. Does the use of additional data from a related language, Greenlandic (KL), improve the outcome? Due to the scarcity of EN–IU parallel data, we investigate if adding Greenlandic data to the Inuktitut data to train a multilingual NMT system (Johnson et al., 2017), improves the performance of the NMT systems on the unconstrained task (Zoph et al., 2016).
- RQ3. Does the translation benefit from using contextual word embeddings? The use of such embeddings has proven beneficial for many tasks in natural language processing (Devlin et al., 2019), including MT (Zhu et al., 2020), so we deem it sensible to test this for a polysynthetic language, which we will do by means of masked language modelling pre-training.

In section 2 we present the main data and evaluation measures used. In section 3 we present experiments with morphological segmentation methods. Section 4 presents the results of our translation systems, and in section 5 we present our conclusions.

2 Corpora and Evaluation

The preprocessing followed the procedure of Joanis et al. (2020), carrying out the following steps in order: spelling normalisation and romanisation (only for IU), punctuation normalisation, tokenisation, and truecasing (only for EN). Parallel data is additionally filtered (ratio 15, minimum and maximum length 1 and 200, respectively). As monolingual data we use the Common Crawl (CC) corpus for Inuktitut, and the 2019 version of Newscrawl for English. For CC we also filter out duplicate lines, lines of which more than 10% of the characters are neither alphanumerical nor standard punctuation, and lines that contain more than 200 words. These

[1] http://www.statmt.org/wmt20/translation-task.html

Proceedings of the 5th Conference on Machine Translation (WMT), pages 274–281
Online, November 19–20, 2020.

steps reduce the amount of data considerably, from 164,766 to 28,391 lines. Line deduplication is also applied to Hansards.[2]

Since the parallel training data contains only Hansards, we used part of the news from the dev set as additional training data by splitting the news part of the dev set: the first 1859 lines are used for training and the last 567 for development. We refer to these subsets as `newsdevtrain` and `newsdevdev`, respectively.

Tables 1 and 2 show the parallel and monolingual datasets, respectively, used for training after preprocessing.

		Words	
Corpus	**Sentences**	**EN**	**IU**
Hansards	769810	17303903	8236210
Newsdevtrain	1859	40154	24121

Table 1: Preprocessed EN–IU parallel training data.

Lang.	**Corpus**	**Sentences**	**Words**
IU	Common Crawl	28391	381805
EN	Newscrawl	5000000	143776337

Table 2: Preprocessed monolingual training data.

During development, we evaluated our systems on the news and Hansards portions of the development set, separately. We used two automatic evaluation metrics: BLEU (Papineni et al., 2002) and CHRF (Popović, 2015). CHRF is our primary evaluation metric for EN→IU, due to the fact that this metric has been shown to correlate better than BLEU with human evaluation when the target language is agglutinative (Bojar et al., 2016). BLEU is our primary evaluation metric for IU→EN systems, as the correlations with human evaluation of BLEU and CHRF are roughly on par for EN as the target language. Prior to evaluation the MT output is detruecased (only EN) and detokenized with Moses' scripts.

3 Segmentation with intrinsic evaluation

Like many polysynthetic languages, Inuktitut has a high degree of inflection and agglutination, leading to very long words with a very high morpheme-to-word ratio (Mager et al., 2018). By our estimation, Inuktitut has an average of around 4.39 morphemes per word.

This means on average there are more potential boundaries, as well as more actual segmentation boundaries to locate per word, making segmentation particularly challenging.

Inconsistent segmentation harms an NMT model's ability to extract knowledge, because it reduces the frequency and activation of all vocabulary items during training, such that for each individual element in the vocabulary is found in fewer contexts. At inference, inconsistent segmentation can result in morphs that are out-of-vocabulary, resulting in information loss.

We hypothesize that linguistically correct segmentation may be particularly beneficial for translation with polysynthetic languages because it could provide more consistent isolation of concepts into subwords.

We evaluated a broad pool of segmenters to determine how close various methods can achieve linguistically correct segmentation, comparing results to reference segmentations obtained from the Inuktitut Computing GitHub repository[3]. This repository contains 1096 Inuktitut words, manually segmented at the National Research Council of Canada (NRC).

Our experiments include: Rule-based with Uqailaut[4]; Morfessor Baseline (semi-supervised) (Creutz and Lagus, 2002); Morfessor FlatCat (semi-supervised) (Grönroos et al., 2014); LMVR (unsupervised) (Ataman et al., 2017); and Neural Transformer segmentation (supervised).

We used Uqailaut's rule-based segmenter to create additional annotated segmentations used to train the supervised and semi-supervised systems. In total 600,000 segmentations of unique words from the Hansard training dataset were created. All semi-supervised and unsupervised systems were trained with the Hansard training corpus. For training semi-supervised methods, we use 60,000 of the collected segmentations with Uqailaut as annotated training data, and another 3,000 as validation data. For LMVR we set the maximum lexicon size to 20,000.

Related to our work, a previous study (Kann et al., 2018) compared segmentation methods based on their ability to generate linguistically correct segmentations for several low-resource Mexican polysynthetic languages. Their proposed RNN-

[2]We used Hansards for training with and without deduplication and the former led to better results.

[3]https://github.com/LowResourceLanguages/InuktitutComputing

[4]http://www.inuktitutcomputing.ca/Uqailaut/info.php

based neural approach outperformed baselines of other common approaches, so we also tested a neural segmentation method, but instead of an RNN we use a Transformer architecture. We implement this neural segmenter using Marian[5]. On the source side, the unsegmented words are used as input data. The corresponding segmented words are used as target data. On the target side we denote the segmentation boundary by adding a boundary token (@), like in the following example:

Source: a k i r a q t u q t u t

Target: a k i r a q @ t u q @ t u t

We trained three neural segmentation models: one on all 600,000 annotated segmentations, plus two with 45,000 annotated segmentations, one with only unambiguous annotations[6] and one with a random selection from the pool of 600,000.

Table 3 shows the intrinsic evaluation results. Similar to Kann et al. (2018), the neural segmentation model improves over existing segmentation methods by a considerable margin. The neural model trained on the 45,000 unambiguous data outperformed the model trained on all the 600,000 segmentations, suggesting that the consistency of the data is more important than the quantity. The other segmenters clearly struggled with the long words, often splitting words into a combination of very long root, and very short morphs. FlatCat scored the highest of the existing methods on both F1 and accuracy.

Unfortunately, both the neural and rule-based models sometimes fail to segment the input word. This makes them unfit to use in a translation system; since some words are left unsegmented, and this leads to a very large vocabulary size which hurts the translation performance. Micher (2017) previously explored improving the coverage of the Uqailaut morphological analyser with the use of an RNN based approach. In Micher (2018), an SRNN extension to the Uqailaut morphological analyzer is used in an SMT system, and yields a statistically significant improvement for IU→EN translation compared to the unextended rule-based analysis. Similar to their approach, we combined the best performing models of the intrinsic evaluation, to construct a custom 3-step segmenter to improve the coverage. This method initially applies the rule-based segmenter. If the rule-based segmenter fails, it falls back on the Transformer (unambiguous 45K) model. For non-alphabetic tokens we apply the BPE 5K model, because the Transformer fails for these tokens.

[5]https://marian-nmt.github.io/

[6]Out of the 600,000 words, Uqailaut produces unambiguous segmentations for 45,000 words

Method	F1	Acc.	Fail (%)
M. Baseline	0.317	0.222	-
M. FlatCat	0.397	0.328	-
LMVR	0.296	0.240	-
Trf. (45K rand.)	0.378	0.297	-
Trf. (45K single)	**0.680**	**0.539**	0.09
Trf. (all 600K)	0.625	0.433	0.55
3-Step	0.741	0.696	-
3-Step + LMVR	0.292	0.258	-
Rule-based	0.716	0.681	11.50

Table 3: Results of the intrinsic evaluation for each segmentation approach. The F1 score is calculated on segmentation boundaries, while the accuracy is calculated on the full segmentation. The *fail* statistic signifies the percentage of words that the approach failed to reconstruct for the methods for which that can occur.

Preliminary experiments with this approach still resulted in a very large vocabulary size. To reduce the vocabulary size further and combine all steps into a single model, afterwards we perform vocabulary reduction using LMVR. We specify a lexicon size of 20,000, which results in an actual vocabulary size of 41,024. The vocabulary reduction applied to the 3-step model leads to a drop in F1 and accuracy. This could be either because the vocabulary reduction leads to fewer segmentation boundaries per word, or because LMVR changes the model too much.

4 Translation experiments

Unless mentioned otherwise, the translation models are trained using Marian (Junczys-Dowmunt et al., 2018) v1.9.0 on an Nvidia V100. The translation models use the `transformer` model type with default settings. We use the `ce-mean-words` cost function. We perform a validation run every 5,000 update steps and apply early stopping after the validation cost stalls 5 times in a row. The model with the best translation score on the validation set (Section 2) is stored for each experiment.

4.1 Constrained Systems

Our constrained systems can be divided into four groups according to the techniques used: tags, backtranslation and domain-specific data (section 4.1.1), morphological segmentation (4.1.2),

contextual word embeddings (4.1.3) and ensembling and fine tuning (4.1.4).

4.1.1 Initial Systems

In these systems, following Joanis et al. (2020), we segment the training data with BPE (Sennrich et al., 2016) separately on each language. 5,000 and 2,000 merges are performed on both languages for MT systems into EN and IU, respectively.

Table 4 shows our initial constrained systems and their results on the development set.

	IU→EN		EN→IU	
System	**News**	**Hansards**	**News**	**Hansards**
1	14.73	29.62	40.29	52.97
2	17.96	29.7	47.47	**54.20**
3	17.24	28.88	**51.31**	53.86
4	**22.24**	**30.05**	NA	NA

Table 4: Results of the initial constrained systems for both translation directions and both dev sets. The scores are BLEU (IU→EN) and CHRF (EN→IU). Best result shown in bold.

Initial Systems System 1 is trained on Hansards. System 2 adds `newsdevtrain`, oversampled (5 times) given its small size compared to the other corpus used for training, i.e. Hansards (see Table 1). This results in a notable improvement for news (over 3 points into EN and over 7 into IU) and, as expected, a minor difference for Hansards.

Tags System 3 differs from system 2 in that each source sentence is preprended with a tag (<H> for Hansards and <N> for news); this degrades results into EN, but improves results into IU considerably for news (almost 4 points), with minimal change to Hansards.

Backtranslation In system 4 different amounts of newscrawl 2019 were backtranslated and concatenated to the training data of previous systems 3 and 2, with (<B>) and without a tag, respectively. This system is used only for IU→EN and its best results were obtained with 1 million sentences without tags; compared to system 2, adding backtranslation results in over 3 points improvement for news (22.2 vs 18) and a smaller increase for Hansards (30 vs 29.7).

We also explored the use of backtranslation for EN→IU. CC (backtranslated into EN) was concatenated to the training data of the previous systems 3 and 2, with and without a tag, respectively. Results were very similar. We conjecture this was due to its limited size and noisy nature, since it is web crawled.

	IU→EN		EN→IU	
Model	**News**	**Hans.**	**News**	**Hans.**
BPE 5K	14.77	**28.31**	32.52	39.81
Morfessor	13.39	26.82	28.75	38.20
FlatCat	12.86	26.49	23.25	29.88
LMVR	14.98	27.50	**34.84**	**41.25**
Trf. (single)	11.31	24.56	31.34	39.33
3-St.+LMVR	**15.25**	28.06	34.51	40.54

Table 5: Results of the extrinsic evaluation for the selected segmentation methods. Scores for IU→EN are in BLEU, and for EN→IU are in CHRF. Best results for each dataset and metric are in bold. All models are trained only on the Hansard training data.

Topic-specific News Because the texts in both dev sets concern (mostly) events in Nunavut, we hypothesised that Nunavut-related news *only* from our backtranslated news might be beneficial. We selected only documents from the document-delimited version of `newscrawl` that contain any word from a topic list.[7] Topic words were picked due to being frequent in `newsdevtrain` and unambiguously related to Nunavut. 2,845 newsstories were extracted, after preprocessing 150,472 sentences and 3,220,925 words. We trained systems with this topic-specific backtranslated news as well as a similar amount of news randomly selected. Contrary to our hypothesis, the random news outperformed topic-specific news: 18.92 vs 20.2 BLEU on the news part of the dev set.

4.1.2 Morphological segmentation

We train translation models for the segmentation methods described in Section 3. For these experiments, the English data was segmented using BPE with 5,000 merges. Results are reported in Table 5. Both models that use LMVR for vocabulary reduction perform well for translation into IU, outperforming BPE on both Hansard and News data. There seems to be no benefit from the use of a more morphologically correct segmenter, as the highest scoring segmenters on the intrinsic evaluation (Table 3) generally performed worse on the extrinsic evaluation.

Based on the results of this extrinsic evaluation, we decide to use the BPE, LMVR, and 3-Step seg-

[7] Baffinland, Inuit, Inuits, inuits, Inuktut, Inuktitut, Iqaluit, Kivalliq, Nunatsiaq, Nunavik, Nunavut and Savikataaq.

mentations in our best systems so far (system 3 into IU and 4 into EN, see Table 4). Different amounts of BPE merges were tried for EN. The best results were obtained with 32,000 into IU and 20,000 into EN, whose results are reported in Table 6. The LMVR segmenter improved the translation into IU for the Hansard data, but not for news. For translation into EN there was no improvement from using a different segmenter.

	IU→EN		EN→IU	
System	**News**	**Hans.**	**News**	**Hans.**
Sys. 4 & 3 resp.	**22.24**	**30.05**	**51.31**	53.86
LMVR	21.89	29.20	50.36	**54.45**
3-Step + LMVR	21.79	29.66	50.19	52.18

Table 6: Results of the constrained systems that use morphological segmentation for both translation directions and both dev sets. The scores are BLEU (IU→EN) and CHRF (EN→IU). Best results shown in bold. The IU→EN models are based on system 4, while the EN→IU models are based on system 3 (Section 4.1.1).

4.1.3 Contextual Word Embeddings

With the recent success of pretrained contextual embeddings in MT (Lample and Conneau, 2019; Zhu et al., 2020), we try using this technique for a polysynthetic language. Specifically, we use the XLM model (Lample and Conneau, 2019), not only as a means of having contextual embeddings, but also to leverage available monolingual data for the task. For our XLM experiments, pretraining uses both masked language modeling (MLM) and translation language modeling (TLM). For the NMT training step, we include both denoising and back-translation for the monolingual data, as well as the standard MT training with the parallel data. Both the pretraining step and the NMT step use the monolingual data and the parallel data.

Pretraining	**IU→EN**	**EN→IU**
No	**19.32**	48.36
Yes	18.58	**49.10**

Table 7: Comparison of pretrained and non-pretrained XLM systems on the News dev set. The scores are BLEU (IU→EN) and CHRF (EN→IU).

To observe the effect of language model pretraining, we train a model using the same data used in system 4 (see Table 4), with 10,000 BPE joins applied jointly to both languages.[8] See results in Table 7. Interestingly, the performance decreases for IU→EN but increases for EN→IU when pretraining is added. A possible explanation for this is that Inuktitut stands to benefit more from pretraining as it uses more of the total joint vocabulary (around 90% of the tokens compared to 70%).

To use the existing monolingual data (Section 2), we train XLM models with the News Crawl data for English and Common Crawl data for Inuktitut, as specified in Table 2. We also use Hansards and `Newsdevtrain` oversampled 5 times for parallel data. We try both tagging the data (with the Common Crawl data receiving its own tag, <C>) and leaving it untagged. We report the results in Table 8. The results indicate an improvement with tagged data in the EN→IU direction. This is consistent with our observations with Marian-run models (systems 2 and 3 in Table 4). The XLM model results show that despite removing back-translated parallel data, results are similar. This is almost certainly due to the on-the-fly back-translation present in the training scheme. The results for EN→IU are improved, which is likely due to even a small amount of Inuktitut Common Crawl data being indeed useful for training.

Tagged	**IU→EN**	**EN→IU**
No	**18.96**	48.9
Yes	16.76	**49.97**

Table 8: Results of the XLM models using monolingual data on the News dev set. Scores are BLEU (IU→EN) and CHRF (EN→IU).

The best result with XLM (19.32 BLEU for IU→EN) is almost 3 points behind the result of the system trained with Marian on the same data (22.24, system 5 in Table 4). A difference between these two systems is that XLM uses joint BPE (since the encoder is shared by both languages), while with Marian we used separate BPE models for each language, following Joanis et al. (2020). To have a fairer comparison, we train the same Marian model with joint BPE, which leads to a score of 21.43, still 2 points ahead of the XLM model.

This difference in performance can be attributed, we hypothesise, to two reasons: (i) the XLM models use a joint encoder and decoder for both languages so the model must learn to translate in both

[8] We apply BPE jointly as it follows the methods of Lample and Conneau (2019).

directions and (ii) differences in implementation of the Transformer model in both toolkits.

4.1.4 Ensembles

For our final submissions, we depart from the best system so far (3 into IU and 4 into EN) and experiment with the use of ensembling and fine-tuning techniques. While some systems that used morphological segmentation performed similarly to those with BPE, their ensembles lagged behind. We therefore focused on BPE-based systems. In the following experiments we varied the value of the decoder's penalty length based on results on the dev set (until now we had used the value 1.0): for IU→EN we use 0.8 for news and 1.4 for Hansards while for EN→IU 1.2 was used for both dev sets. The results are shown in Table 9.

	IU→EN		EN→IU	
System	**News**	**Hans.**	**News**	**Hans.**
best single system	22.38	38.41	51.83	54.35
ens normal	23.72	39.07	52.92	55.05
ens FT	24.01	**39.72**	53.19	55.31
ens normal + FT	**24.25**	39.67	**53.46**	**55.39**

Table 9: Results of the constrained systems that use ensembling (referred to as ens) and fine tuning (FT) for both translation directions and both dev sets. The scores are BLEU (IU→EN) and CHRF (EN→IU). Best results shown in bold.

Ensembles are built by training the same system with different seeds (4 into EN and 3 into IU) and picking the model from each training seed with the highest score. These bring consistent improvements for both directions and dev sets: from 0.66 points for IU→EN Hansards to 1.34 for news in the same direction (row "ens normal" in Table 9).

We fine tune on `newsdevtrain` on its own and together with backtranslated news (only into EN) for the news dev set and on Hansards for the Hansards dev set. The ensembles of fine-tuned models bring consistent improvements compared to ensembles of non fine-tuned systems (row "ens FT" versus "ens normal" in Table 9). Finally, ensembling both fine-tuned and no fine-tuned systems (row "ens normal + FT" in Table 9) pushes the scores further (except for Hansards IU→EN) though rather slightly.

4.2 Unconstrained Systems

4.2.1 Data Acquisition

We use three additional parallel corpora that we acquired. First, we use data from the Inuktitut magazine[9], which contains parallel articles about Inuit culture and society in Inuktitut (IU), English (EN), and French; we manually extracted the text (IU syllabics, romanized IU, and EN) from several recent issues. Second, we use data from a Kalaallisut (KL) magazine[10] containing parallel news articles in Danish (DA) and KL. These texts were also manually extracted. Thirdly, parallel data from 21 multilingual websites containing DA and KL texts, was crawled using bitextor[11].

4.2.2 MT with Unconstrained Data

These datasets are pre-processed just like the ones from the constrained setup. In addition, we select a subset using their sentence alignment confidence score.[12] The KL crawl is paired with Danish. We performed language classification on the Danish data using LangID[13], removing any sentence pairs not classified as Danish. Danish was translated into English with a pretrained DA→EN system[14] from OPUS-MT (Tiedemann and Thottingal, 2020). Dataset details are presented in Table 10.

		Words	
Corpus	**Sentences**	**EN**	**IU/KL**
IU Magazine	1134	29312	18152
KL Magazine	657	13009	7491
KL crawl	14778	277159	163468

Table 10: Preprocessed unconstrained parallel training data.

We added these corpora atop the best constrained systems (3 into IU and 4 into EN) one at a time and evaluated on the news part of the dev set. Table 11 shows the results. Into EN, adding IU magazine (for which we tried different oversampling values) did not improve results. Due to this and time limitations we did not add the remaining unconstrained

[9] *Inuktitut Magazine*, https://www.itk.ca/category/inuktitut-magazine/.

[10] *Atuagagdliutit*, https://timarit.is

[11] https://github.com/bitextor/bitextor

[12] The datasets were aligned with Hunalign, which provides a confidence score. We experimented with different thresholds and based on results on the dev set and used 0.4 for IU and KL magazines and 0.5 for KL crawl (Varga et al., 2007).

[13] https://github.com/saffsd/langid.py

[14] https://object.pouta.csc.fi/OPUS-MT-models/da-en/opus-2019-12-04.zip

data. Into IU, adding IU magazine (with a tag and oversampled 5 times) resulted in a slight improvement (51.9 vs 51.3). Adding to this KL magazine (also oversampled 5 times) degraded results, as did adding KL crawl (although to a lesser extent).

System	IU→EN	EN→IU
Best constrained (5, 3 resp.)	**22.24**	51.31
+ IU magazine	22.22	**51.88**
+ IU mag + KL mag		50.57
+ IU mag + KL crawl		51.27

Table 11: Results of the unconstrained systems for both translation directions and both dev sets. The scores are BLEU (IU→EN) and CHRF (EN→IU). Best results shown in bold.

5 Conclusions

This paper has reported on the systems submitted by the University of Groningen to the English↔Inuktitut translation directions of the news shared task at WMT 2020.[15] Our best results were obtained using well-established techniques, including oversampling domain-specific training data, backtranslation, tags, fine-tuning and ensembling.

The use of morphological segmentation (RQ1) led to results that were on par with those obtained by BPE in terms of automatic evaluation metrics. One problem is that existing morphological segmenters for low-resourced languages like Inuktitut suffer from poor coverage, which impedes making a complete comparison with more automatic methods. The extrinsic comparisons between segmenters showed that a more accurate morphological segmentation does not lead to improved translation performance. We further found that existing language agnostic segmenters struggle to produce correct segmentations on Inuktitut, and that neural methods appear to be more suitable for polysynthetic languages (cf. (Kann et al., 2018)) . Note also the importance of limiting the vocabulary size of morphological segmentation for MT, which could be explored further.

The use of additional data from Inuktitut did improve the results slightly, but not the addition of data from a related language, Greenlandic (RQ2). The fact that its usefulness was limited could be due to the fact that half of the test set was from a specific domain for which considerable amounts of data were already available to train (Hansards).

Finally, the use of contextual embeddings (RQ3), led to mixed results since it resulted in an improvement for one direction but a degradation for the other.

[15] We will provide links to the additional datasets we used in the camera-ready version.

Acknowledgments

We would like to thank the Center for Information Technology of the University of Groningen for their support and for providing access to the Peregrine high performance computing cluster. Thanks also to Ben Shaffrey, Barbera de Mol and Adna Bliek for help preparing the Inuktitut magazine data.

References

Duygu Ataman, Matteo Negri, Marco Turchi, and Marcello Federico. 2017. Linguistically motivated vocabulary reduction for neural machine translation from turkish to english. *The Prague Bulletin of Mathematical Linguistics*, 108(1):331–342.

Ondřej Bojar, Yvette Graham, Amir Kamran, and Miloš Stanojević. 2016. Results of the WMT16 metrics shared task. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 199–231, Berlin, Germany. Association for Computational Linguistics.

Mathias Creutz and Krista Lagus. 2002. Unsupervised discovery of morphemes. In *Proceedings of the ACL-02 Workshop on Morphological and Phonological Learning*, pages 21–30. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Stig-Arne Grönroos, Sami Virpioja, Peter Smit, and Mikko Kurimo. 2014. Morfessor flatcat: An hmm-based method for unsupervised and semi-supervised learning of morphology. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1177–1185.

Eric Joanis, Rebecca Knowles, Roland Kuhn, Samuel Larkin, Patrick Littell, Chi-kiu Lo, Darlene Stewart, and Jeffrey Micher. 2020. The Nunavut hansard Inuktitut–English parallel corpus 3.0 with preliminary machine translation results. In *Proceedings of*

The 12th Language Resources and Evaluation Conference, pages 2562–2572, Marseille, France. European Language Resources Association.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia. Association for Computational Linguistics.

Katharina Kann, Jesus Manuel Mager Hois, Ivan Vladimir Meza-Ruiz, and Hinrich Schütze. 2018. Fortification of neural morphological segmentation models for polysynthetic minimal-resource languages. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 47–57, New Orleans, Louisiana. Association for Computational Linguistics.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *Advances in Neural Information Processing Systems (NeurIPS)*.

Manuel Mager, Elisabeth Mager, Alfonso Medina-Urrea, Ivan Vladimir Meza Ruiz, and Katharina Kann. 2018. Lost in translation: Analysis of information loss during machine translation between polysynthetic and fusional languages. In *Proceedings of the Workshop on Computational Modeling of Polysynthetic Languages*, pages 73–83, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Jeffrey Micher. 2017. Improving coverage of an Inuktitut morphological analyzer using a segmental recurrent neural network. In *Proceedings of the 2nd Workshop on the Use of Computational Methods in the Study of Endangered Languages*, pages 101–106, Honolulu. Association for Computational Linguistics.

Jeffrey Micher. 2018. Using the Nunavut hansard data for experiments in morphological analysis and machine translation. In *Proceedings of the Workshop on Computational Modeling of Polysynthetic Languages*, pages 65–72, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Maja Popović. 2015. chrf: character n-gram f-score for automatic mt evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395.

Víctor M. Sánchez-Cartagena and Antonio Toral. 2016. Abu-MaTran at WMT 2016 translation task: Deep learning, morphological segmentation and tuning on character sequences. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 362–370, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Jörg Tiedemann and Santhosh Thottingal. 2020. OPUS-MT — Building open translation services for the World. In *Proceedings of the 22nd Annual Conferenec of the European Association for Machine Translation (EAMT)*, Lisbon, Portugal.

Dániel Varga, Péter Halácsy, András Kornai, Viktor Nagy, László Németh, and Viktor Trón. 2007. Parallel corpora for medium density languages. *Amsterdam Studies In The Theory And History Of Linguistic Science Series 4*, 292:247.

Jinhua Zhu, Yingce Xia, Lijun Wu, Di He, Tao Qin, Wengang Zhou, Houqiang Li, and Tie-Yan Liu. 2020. Incorporating bert into neural machine translation.

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer learning for low-resource neural machine translation. *arXiv preprint arXiv:1604.02201*.

OPPO's Machine Translation Systems for WMT20

Tingxun Shi, Shiyu Zhao, Xiaopu Li, Xiaoxue Wang, Qian Zhang
Di Ai, Dawei Dang, Zhengshan Xue and **Jie Hao**
Manifold Lab, OPPO Research Institute, Beijing, China
{shitingxun, zhaoshiyu, lixiaopu, wangxiaoxue, zhangqian666,
aidi1, dangdawei, xuezhengshan, haojie}@oppo.com

Abstract

In this paper we demonstrate our (OPPO's) machine translation systems for the WMT20 Shared Task on News Translation for all the 22 language pairs. We will give an overview of the common aspects across all the systems firstly, including two parts: the data preprocessing part will show how the data are preprocessed and filtered, and the system part will show our models architecture and the techniques we followed. Detailed information, such as training hyperparameters and the results generated by each technique will be depicted in the corresponding subsections. Our final submissions ranked top in 6 directions (English ↔ Czech, English ↔ Russian, French → German and Tamil → English), third in 2 directions (English → German, English → Japanese), and fourth in 2 directions (English → Pashto and and English → Tamil).

1 Introduction

This paper describes the OPPO's submission to the Fifth Conference on Machine Translation (WMT20) news translation shared task. We built Transformer (Vaswani et al., 2017)-based systems for all the directions, and applied several well-known, widely-used techniques, such as large-scale back-translation (Sennrich et al., 2016a) and forward-translation, model ensemble and reranking. Since all the systems share a roughly similar data preprocessing and training methods, to avoid duplication words, we will demonstrate the common knowledge in Section 2 firstly, which will be divided into two parts: the preprocessing part shows the data preprocessing pipeline and data filtering pipeline, the latter is generally composed by rule-based filtering and alignment-based filtering; the training part depicts the techniques we applied. Detailed information, including training hyperparameters, the results generated by each technique, and some other explorations will be listed in each corresponding direction in Section 3. Finally, we will summarize the report and indicate our final works. We used *marian* (Junczys-Dowmunt et al., 2018) to implement our systems for English ↔ {Khmer, Russian, Tamil} and French ↔ German task pairs, and *fairseq* (Ott et al., 2019) for the rest [1].

2 System Overview

We preprocess corpora in two stages. In the preprocessing stage, data is converted but not filtered. The common pipeline of preprocessing including the following steps:

- Remove non-utf8 characters
- Unescape html characters, e.g. "`>`" is converted to ">"
- Normalize different kinds of spaces and punctuations
- Tokenization
- True case

The last three steps are all processed by *moses* scripts. This pipeline is both applied for the parallel corpora and monolingual corpora, and true case models are generally trained on the mixture of parallel and monolingual datasets.

After preprocessing we filter the parallel corpora according to statistical information and alignment information, set the thresholds according to our previous experiences. For the statistic perspective, we mainly focus on some heuristic rules, contain but not limited in

[1] Choice on the training framework is only depends on personal habit.

Proceedings of the 5th Conference on Machine Translation (WMT), pages 282–292
Online, November 19–20, 2020.

- Pairs of which the source side and the target side are the same.
- Pairs contain blank lines.
- Pairs contain too long sentences (typically those have more than 200 words).
- Pairs that have abnormal source-target length ratios. The source-target length ratio is defined as the words count ratio between the source and the target. Typically the upper bound is 2.5 and the lower bound is 0.4.
- Pairs that have irregular character-word length ratios. The character-word length ratio is defined as the ratio between the count of characters and the count of words. Generally the upper bound is 12 and the lower bound is 1.5.
- Pairs that contain too long words. The length threshold for deciding whether the word is too long is 25 characters.

For the alignment perspective, we use *fast_align* (Dyer et al., 2013) to acquire the alignment scores from source to target and vice versa, then we average the scores for each pair to calculate a data pair's sentence-level alignment score. If a sentence pair's sentence-level alignment score is lower than -15, it will be expelled from the final dataset.

Having purified the corpus, we generally try to boost our systems using the following techniques, step by step:

1. Back-translation and forward-translation. Using the trained models to translate big volume, monolingual corpus from the target side to source side (i.e. back-translation (Sennrich et al., 2016a)) has been proved a very successful method in the past practices. In our experiments we can also see a general improvement brought by this technique, but it is not always the case. We also tried sampling based back-translation proposed in (Edunov et al., 2018), and this is effective only in certain cases as well. Furthermore, we found translating from the monolingal corpus from source language can also bring gains for the models (consistent with the phenomenon depicted in (Burlot and Yvon, 2018)), but in this situation arg-max based beam search should always be applied. We also followed (Hoang et al., 2018) to iteratively back-translate and forward-translate the corpus for several times.
2. Fine-tune. Adding too many synthetic parallel data generated by machine translation models could potentially modify the latent data distribution, and in some tasks the provided monolingual dataset has a small difference from the required domain (news), so after having trained models from the mixture of the original parallel corpus and the synthetic dataset, we continue fine-tune our models on the original parallel datasets only. Besides, for some low-resource tasks (such as tasks on Pashto and Khmer), even the official training datasets have relatively lower qualities, therefore only using training dataset to fine-tune is still not enough. For these tasks, we took one more step to fine-tune the models on the official released validation set, and we can always see a further improvement.
3. Ensemble. We generally train and fine-tune several different models and compose them into an ensemble model for a better result.
4. Reranking. With the ensemble model in the hand, we usually generate k-best candidates and use different scorers to score them. Scorers can be divided into three groups: **forward scorers** are just another ensemble models composed by the forward translation models (models translate the source language to the target language). Suppose we have trained 6 base forward models, typically we compose all of them together to form a big ensemble model for generating final results (this model is also used as a scorer), and then additionally enumerate all the 5-combinations of them to get another $\binom{6}{5} = 5$ scorers. Sometimes we furthermore enumerate all the 4-combination to get $\binom{6}{4} = 15$ more scorers for better reranking. **backward scorers** are ensemble models that actually back-translation models (models translate the target language to the source language), and **language models** are ensemble language models of target language. For each group of the scorers, we may use the left-to-right (l2r) models or right-to-left (r2l) models. For the latter form, we reverse the words orders for both source sentences and target sentences and train the models. The scores

generated by those scorers are used as features by the reranking model. For reranking, we mostly applied K-Batched MIRA (Cherry and Foster, 2012) or noisy channel (Yee et al., 2019).

3 Experiments Details

In this section we demonstrate our experiments details for each direction. For brevity we will ignore the same preprocessing and techniques we introduced in the previous section, mainly focus on how the techniques boosted the systems, and some other unique observations we found during the experiments.

In the text we will sometimes use ISO-639-1 two-letter codes for each language for short. Mapping between the abbreviations and full names can be found in Table 1. For example, when talking about the English → Chinese task, we may write `EnZh` for short, capitalizing the first letter of the ISO-639-1 codes for both source languages and target languages. For the direction pairs that involve English, sometimes we use the non-English language to indicate the whole pair, e.g. "Russian tasks" is used to indict the English ↔ Russian bidirectional task. As this report is in the news task scope, we sometimes use "task" as a synonym of "direction", e.g. "`EnZh` task" means the direction that translates English to Chinese.

By default, for every sub-task we combine all the official provided parallel corpora into a big dataset then clean it, use the cleaned corpus to train our baseline models. We strictly followed the requirement of the contest to use official released datasets only, so the systems we built are all constrained systems. If not mentioned, all of our baseline models are trained on the parallel corpus only, and all the scores reported are calculated by sacreBLEU (Post, 2018) based on the results which has been removed BPE symbols, detruecased and detokenized. We always apply BPE subwords (Sennrich et al., 2016b) on the corpora, usually train Transformer-Big models and tie the input and output matrices of the decoder. For all the tasks, we used Adam optimizer (Kingma and Ba, 2014). All the main systems (i.e. submitted results) are generated by the model listed in the **last** row of the corresponding table in each task.

Language Name	ISO-639-1 Code
Chinese	`zh`
Czech	`cs`
English	`en`
French	`fr`
German	`de`
Inuktitut	`iu`
Japanese	`ja`
Khmer	`km`
Pashto	`ps`
Polish	`pl`
Russian	`ru`
Tamil	`ta`

Table 1: ISO-639-1 codes for languages appear in news task of WMT20

3.1 English ↔ Chinese

3.1.1 Data Preprocessing

Compared from the other languages in the shared task, especially the languages which use alphabetical writing systems, Chinese has three typical characteristics, leading to three extra preprocessing steps we introduce below:

1. Chinese has two different writing systems: simplified Chinese and traditional Chinese. Following the statistical information mined from the original parallel corpus, we converted all traditional Chinese characters to their simplified counterparts.

2. Some websites use GB2312 to encode texts, therefore could convert Latin letters, digit characters and some other punctuation marks into *full width* form. Besides of some particular punctuation marks (full stops, commas, question marks and exclamation marks), we converted all the other symbols to half width form.

3. Chinese does not have explicit words boundaries, all the characters in the same clause are connected together. We used *pkuseg* (Luo et al., 2019) to segment words from the text.

It should be noted that Japanese also has these three features, so the same process is also applied in the English ↔ Japanese systems.

For data filtering stage, besides the heuristic rules we demonstrated in the previous section, we also compare the count of numbers and punctuation marks between source side and target side. If

the difference on number counts is greater than 3 or the difference on punctuation marks counts is greater than 5, the sentence pairs will also be removed.

3.1.2 Training

We combined the Chinese corpus and English corpus together to train BPE. The total BPE operation merge counts is 36K. After learning BPE operations, we built vocabularies for each language separately. The final vocabulary size for Chinese is 42K and for English is 23K. The model architecture for both directions are all Transformer-big. For `ZhEn` task, we tried different hyperparameters to train several models for getting ensemble model: learning rates ranged from 0.0003 to 0.0008, warmup steps fixed at 16,000, dropout ranged from 0.2 to 0.3. For `EnZh` task, the hyperparameters are all fixed (but tried different random seeds): learning rate was 0.0003, warmup steps was 15,000, feed forward network dimension was 15,000.

Entity substitution is experimented in the `ZhEn` system. We use *StanfordNLP* (Qi et al., 2018) to do the NER from parallel corpus and Chinese monolingual datasets (Because in Chinese monolingual datasets an annotation usually follows a foreign name). After having extracted all the entities, we didn't use alignment information to build the mapping between Chinese entities and English entities, but constructed such relationship just according to co-occurrence frequency information: suppose an entity "北京" occurs 50 times totally in the Chinese corpus from 20 sentences, and in the corresponding 20 English sentences "Beijing" occurs 51 times, "Shanghai" occurs 10 times, then we believe "北京" can be translated to "Beijing" but not "Shanghai". With the entity mapping rules, we then replace the entities in the sentence pairs by different tags `<tag1>`, `<tag2>` ... and train models. In the inference time, model generates results with those tags, and we take another post-edit stage to recover the entities, using the mapping rules as lookup tables.

Table 2 shows our systems for `ZhEn` task, and 3 shows our systems for `EnZh` task. For `ZhEn`, we back-translated 20M NewsCrawl and 17M NewsDiscussion monolingual datasets from English to Chinese, and forward-translated 13M Chinese monolingual dataset to English (including XMU, LDC, etc.).

System	BLEU	Improvement
Baseline	28.8	-/-
+ Back-translation	29.8	+1.0/+1.0
+ Forward-translation	34.5	+5.7/+4.7
+ Entity substitution	35.2	+6.4/+0.7
+ Fine-tuned by `newstest2017`	36.7	+7.9/+1.5
+ Ensemble & reranking	38.3	+9.5/+1.6

Table 2: Overview of our WMT20 Chinese → English systems. In the "Improvement" column we report two improvement amounts, the first one is the improvement amount compared with the baseline model (absolute improvement), and the last one is got from comparing with the previous step (relative improvement). Scorers for reranking are composed by 3 forward left-to-right (l2r) models, 3 forward right-to-left (r2l) models, 3 backward r2l models and 2 l2r Transformer language models.

System	BLEU	Improvement
Baseline	38.6	-/-
+ Back-translation (A)	39.1	+0.5/+0.5
+ Fine-tuned by parallel corpus	40.6	+2.0/+1.5
+ Fine-tuned by `newstest2017`	41.3	+2.7/+0.7
+ Forward-translation (B)	41.9	+3.3/+2.8
+ Ensemble	42.7	+4.1/+0.8
+ Reranking	43.2	+4.6/+0.5

Table 3: Overview of our WMT20 English → Chinese systems. BLEU scores are character-level. Model trained by adding forward-translation data (system B) is directly compared with the one trained by adding back-translation data only (system A). The two phases fine-tune, which is effective for the system A, has no obvious impact on system B

3.2 English ↔ Czech

3.2.1 Data Preprocessing

The officially released English ↔ Czech dataset has a different format from the other sub-tasks. The dataset, which is called CzEng 2.0 (Kocmi et al., 2020), contains not only parallel sentence pairs, but also the data source and three scores: alignment score calculated by dual conditional cross-entropy filtering (Junczys-Dowmunt, 2018), and language scores to show of how confident the source is Czech and the target is English. This extra information can further help us to filter the corpus.

Having noticed that both `CsEn` and `EnCs` tasks would be evaluated on long, document-level news datasets, and the CzEng dataset contains some document information, we first analyzed the data sources given in the dataset, to determine which of them are near to the destination domain, and which are far away. The data sources were observed from four aspects: 1. Are the sentences more colloquial or more formal? 2. How well the data is aligned? 3. Can the sentences form a paragraph? 4. Is the corpus also in the news domain?

With the features of the given data sources, we first set a hard condition to check whether a given sentence pair could be kept, then set different probabilities to randomly drop some pairs from certain data sources. Constrained by the paper length we cannot list all of the rules for all the data sources here, but we can take some examples. For the data of which the source is *news*, we kept all of them; at the other extreme, for the *commoncrawl* data, we first removed all the data pairs of which the alignment scores are below than 0.25, or the probabilities of the source sentences belonging to Czech are less than 0.9, then we removed 40% of the remained data randomly.

As the original dataset contains some paragraph information, we concatenated all the sentences that were originally in the same paragraph with a delimiter "|||" (for the sentences that come from the data sources of *subtitles*, *subtitleE* and *subtitleM*, we didn't concatenate them). After the initial filtering, we kept 24.24 million data pairs (If we add in the czeng-test data, the total volume is 24.44 million pairs). The kept data were then processed and filtered by the pipeline presented in the previous section, and we finally got 14.4 million pairs. Detailed preprocessing information can be found in Table 4.

Step	# Sentence pairs kept	Retention rate
Initial filtering	24.44 M	-
Deduplication	17.3 M	70.75%
Heuristic filtering	14.42 M	83.41%
Bad characters filtering	14.40 M	99.84%

Table 4: Preprocessing of the CzEng dataset. Official provided dataset contains alignment information so we didn't calculate alignment scores again, directly reused official information in the initial filtering step. In the "bad characters filtering" step, we printed a character frequency list from the dataset and set a threshold, removed all data pairs that contain irregular characters whose frequencies are lower than the threshold.

System	Score
full-doc	25.7
short-doc	26.3
no-doc	27.0

Table 5: Document-level model training experiments on the `EnCs` task. Scores are reported on `newstest2019` dataset

3.2.2 Model Training

As the evaluation for the `En` ↔ `Cs` tasks would be document-level, we first experimented to see if training a model on a dataset which contains many very long sentences can generate better translations for whole documents. We prepared the datasets in three different ways: 1. Concatenating all sentences that belong to the same document (as indicated in the original data sources), noted as "full-doc"; 2. Concatenating three consecutive sentences together, and select the middle one as the final result from the generated translation, noted as "short-doc"; 3. No special preprocessing, one line contains one sentence, noted as "no-doc". The experiments results are shown in Table 5.

From the results we can find that no extra document related preprocessing is the best preprocessing, so we continued our improvement based on the dataset which does not contain document-level information. We first trained two models based on the full CzEng 2.0 dataset (including all the official translated data). Models are all trained using Transformer-Big architecture with norm clipping set to 0.1, dropout set to 0.3, gradient update frequency set to 8, maximum tokens in a batch set to 6000. Warmup steps and learning rate varied from different experiments, the most common combination is warmup steps set to 16,000 and learning rate set to 0.001. During decoding the beam size is 5 and length penalty is 2.5 for `CsEn`, 2 for

Direction	Dataset	# Data pairs	Score
CsEn	All official released data	122 Million	34.0
CsEn	All official released data + 31M full sampling back-translated data	153 Million	34.1
CsEn	30M data sampled from official released data + 31M full sampling back-translated data	61.2 Million	34.2
EnCs	All official released data	122 Million	28.6
EnCs	All official released data + 28M full sampling back-translated data	150 Million	29.0

Table 6: Models prepared for the final back-translation and forward-translation. Czech monolingual datasets are the combination of all officially provided Newscrawl datasets, English monolingual datasets are sampled from Newscrawl 2019.

EnCs. The score of the CsEn model on offline test set (newstest2018) is 34.0 and the EnCs model on validation set (newstest2019) is 28.6. We use these two models back-translated and forward-translated several data, mixed our synthetic dataset with the original official whole datasets together, and trained several models. Models which have the best performances are selected for the final back-translation and forward-translation, which are listed in Table 6.

We composed the models shown above as two ensemble models, one for each direction, and did another round of back-translation and forward-translation again. For the EnCs task, we prepared two different final datasets as below. Two datasets are all generated by randomness-based back-translation, the difference is the full sampling one sample output words in the full vocabulary, whilst the top-k one restricts the sampling pool in the words that are listed in the top-k highest probabilities for each step:

- **Top-k sampling based dataset**, consists of 24 million data pairs from the original parallel corpus, 54 million officially provided forward-translated corpus (translated from English monolingual corpus), 50 million top-10 sampling back-translated corpus, and 15 million forward-translated corpus generated by our own ensemble model.

- **Full sampling based dataset**, consists of 24 million data pairs from the original parallel corpus, 54 million officially provided forward-translated corpus (translated from English monolingual corpus), 15 million forward-translated corpus generated by our own ensemble model, 31 million "old" full sampling back-translated data used in Table 6, and 36.7 million "new" full sampling back-translated data generated by ensemble model.

System	BLEU	Improvement
Baseline (parallel data only)	27.0	-/-
+ Officially provided synthetic data	28.6	+1.6/+1.6
+ Full sampling based back-translated data	29.0	+2.0/+0.4
+ Ensemble	29.2	+2.2/+0.2
+ FDA fine-tune	29.7	+2.7/+0.5
+ Fine-tune by Newstest2018 & reranking	30.5	+3.5/+0.8

Table 7: Overview of our WMT20 English → Czech systems. Scorers for reranking are composed by 16 forward left-to-right (l2r) models, 3 forward right-to-left (r2l) models, 3 backward r2l models and 3 l2r Transformer language models. We re-learned BPE after adding in the officially provided synthetic data and fixed it for the following steps. The BPE is learned separately and the merge operations count is 36K.

> The 36.7 million "new" back-translated data are generated after an extra cleaning step: As we observed the results generated by full-sampling back-translation sometimes contain very bad sentences, we check how many steps the decoder scores below -10 when decoding for a given input. If 20% of the step scores for a given sentence are below -10, then we discard the sentence pair.

We found the models trained by top-k sampling based dataset are generally worse than those trained by full sampling based dataset, therefore selected one top-k sampling based model and three full sampling based model to form the final ensemble model for decoding the test data. For the CsEn task, The final dataset is composed by 24 million original parallel data pairs, 24 million ensemble knowledge distillation data pairs, 50 million top-k sampling back-translated pairs, 10 million argmax beam search back-translated pairs, and 17 million forward-translated pairs. We trained 4 models using different learning rate (varied from 0.0008 to 0.0015) on this dataset, and fine-tuned them using original parallel dataset (fine-tuning on EnCs models does not bring any gains). The fine-tuned models are used for the final ensemble model. We also applied FDA algorithm (Biçici and Yuret, 2011) on the parallel dataset, picked out 5 million sentence pairs that are similar to the test set and fine-tuned on this small dataset.

The overview of our EnCs system is listed in Table 7, and CsEn system is listed in Table 8

3.3 English ↔ German

For En ↔ De tasks, we generally followed the process depicted in Section 2, cleaned 46.8 million data pairs and kept 30.6 million. For data

System	BLEU	Improvement
Baseline	31.9	-/-
+ Officially provided synthetic data	34.0	+2.1/+2.1
+ Full sampling based back-translated data	34.1	+2.2/+0.1
+ Original parallel data fine-tune	34.8	+2.9/+0.7
+ Ensemble	35.3	+3.4/+0.5
+ FDA fine-tune	35.5	+3.6/+0.2
+ Reranking	35.9	+4.0/+0.4

Table 8: Overview of our WMT20 Czech → English systems. Scorers for reranking are composed by 16 forward left-to-right (l2r) models, 3 forward right-to-left (r2l) models, 3 backward r2l models, 3 l2r Transformer language models and 1 all lower-cased Transformer language model which does not apply BPE on the training dataset.

System	DeEn BLEU	EnDe BLEU
Baseline	40.7 (-/-)	42.6 (-/-)
+ KD	-	44.9 (+2.3/+2.3)
+ Fine-tune on parallel corpus	-	45.3 (+2.7/+0.4)
+ Ensemble	41.9 (+1.2/+1.2)	45.9 (+3.3/+0.6)
+ Reranking	42.2 (+1.5/+0.3)	46.5 (+3.9/+0.6)

Table 9: Overview of our WMT20 German ↔ English systems. Reranking follows noisy-channel reranking (Yee et al., 2019). BLEU scores are reported on `newstest2019`. We learned BPE jointly for both tasks, merge operation is 32K. Learning rate for training is 0.001 and warmup steps is 4000

preprocessing, we removed sentence pairs that contain too many punctuation marks, and too many `[^A-Za-z]` characters. In both directions we found neither back-translation nor forward-translation could yield any gains. In the `EnDe` we found ensemble knowledge distillation (Freitag et al., 2017) could improve the effect but in the `DeEn` task it did not help. The overview of our `En` ↔ `De` system is listed in Table 9.

3.4 English ↔ Inuktitut

We just adapted the official preprocessing script in the syllabic form to process the corpus. BPE was learned independently and the merge operations count is 16K. The overview of our `En` ↔ `Iu` system is listed in Table 10.

System	EnIu BLEU	IuEn BLEU
Baseline	23.7 (-/-)	40.0 (-/-)
+ Back-translation	23.8 (+0.1/+0.1)	40.5 (+0.5/+0.5)
+ Knowledge distillation	-	41.3 (+1.3/+0.8)
+ Ensemble	24.3 (+0.6/+0.5)	41.9 (+1.9/+0.6)
+ Reranking	-	43.7 (+3.7/+1.8)

Table 10: Overview of our WMT20 English ↔ Inuktitut systems. Scores are reported on the official validation set

System	JaEn BLEU	EnJa BLEU
Baseline	22.0 (-/-)	37.0 (-/-)
+ Back-translation	24.5 (+2.5/+2.5)	41.4 (+4.4/+4.4)
+ Knowledge distillation	25.1 (+3.1/+0.6)	41.4 (+4.4/+0.0)
+ Ensemble	25.7 (+3.7/+0.6)	42.1 (+5.1/+0.7)
+ Reranking	26.1 (4.1/+0.4)	42.5 (+5.5/+0.4)

Table 11: Overview of our WMT20 English ↔ Japanese systems. Reranking follows noisy-channel reranking (Yee et al., 2019). BLEU scores are reported on the offline official validation set, for `EnJa`, we report the character-level score. We trained BPE separately for both tasks, merge operations is 32K. Learning rate for training is 0.0003 and warmup steps is 15000. We tried two different feed forward network dimensions, 4096 and 15000, and found no big differences

3.5 English ↔ Japanese

Our `En` ↔ `Ja` systems generally follow our `En` ↔ `Zh` systems depicted before, the difference was the upper bound of sentence length limit was set to 180 words, and we also set the lower bound to 3. For Japanese word segmentation we used *mecab* [2]. We cleaned 17.64 million parallel pairs and 13.7 million left. For back-translation, we used 16 million Japanese monolingual data and 13 million English monolingual data. The overview of our `En` ↔ `Ja` system is listed in Table 11

We tried to fine-tune the models using original parallel dataset, but didn't see any gain. After the test dataset was released, we applied FDA algorithm and extracted 5000 sentences from the training dataset which are the most similar to the test data. These sentences are mixed with the original validation dataset together, then 500 sentences are split out as a new validation set, the rest were used to fine-tune the models. This step improved our `EnJa` system by 1.3 BLEU and for `JaEn` it is 0.4 BLEU. However, as validation dataset changed and the scores on the new validation dataset were extremely high, this step is not listed in the Table 11.

3.6 English ↔ Khmer

For the Khmer tasks (and some other tasks in the following), The data preprocessing stages are slightly different from the way we depicted in the second section, stricter in the filtering part, which would remove the sentence pair if...

1. It is a duplicated example
2. The source or target side is empty

[2] `https://taku910.github.io/mecab/`

3. It contains urls

4. It has words that contain more than 4 consecutive repeated characters

5. It has unpaired quotation marks or parentheses (not applicable for Khmer tasks, but applied in the other tasks shown later)

6. The punctuation marks between the source and the target cannot be matched (not applicable for Khmer tasks, but applied in the other tasks shown later)

7. The length ratio between the source and target is greater than 2.0 or less than 0.5 (for Khmer is between 0.33 and 3)

8. More than half of the tokens are not from the indicated language. We designed a regular expression (noted as regex for short) for each language according to its alphabet, if the word failed to pass the regex, we say it is not from the given language. For example, the regex for English is `[a-zA-Z'-]+`

The maximum sentence length we allowed is also set to 200 words.

Similar to Chinese and Japanese, Khmer does not mark the words boundaries neither, so we used *SEANLP* [3] to do the Khmer word segmentation. After the cleaning, the 4.46 million pairs of sentences had 351K lines left.

It should be noted that the writing system of Khmer, Khmer script, is an *abugida*, means vowels do not have independent symbols, but are stuck after/above/below/in front of the consonants they follow. Roughly, the minimal meaningful unit of Khmer is called Khmer Character Cluster (KCC for short) (Huor et al., 2004), which should be regarded as a whole but actually contains several characters. Original BPE method would break KCC, but this is not what we expect, so we made some modification to keep it (the segmentation tool we used also considered this language feature). We combined Khmer corpus and English to train BPE together, the BPE merge operations count is 8K.

To train the model, we tried different learning rate ranged from 0.0001 to 0.0004, and different warmup steps from 2,000 to 32,000. The overview of our Km ↔ En system is listed in Table 12. Baseline model is trained by Transformer-mini (4-heads Transformer composed by 4 layers, embedding dimension set to 256, feed forward network dimension set to 1024), learning rate ranged from 0.0008 to 0.001, warmup steps fixed at 40,000. For back-translation, we used all officially provided Khmer monolingual data, and 27 million sentences for English from NewsCrawl 2019 and NewsCommentary 2019.

[3] https://github.com/zhaoshiyu/SEANLP

System	KmEn BLEU	EnKm BLEU
Baseline	5.7 (-/-)	2.38 (-/-)
+ Back-translation	13.0 (+7.3/+7.3)	10.15 (+7.77/+7.77)
+ Ensemble	13.6 (+7.9/+0.6)	10.56 (+8.18/+0.41)

Table 12: Overview of our WMT20 Khmer ↔ English systems. We didn't try fine-tune and reranking for these two tasks. BLEU scores are reported on the official offline validation set, reported on the word-level (different from the online character-level evaluation). for EnKm, the score is calculated by *multi-bleu*.

System	EnPs BLEU	PsEn BLEU
Baseline	6.0 (-/-)	12.3 (-/-)
+ Back-translation	10.7 (+4.7/+4.7)	14.5 (+2.2/+2.2)
+ Knowledge distillation	10.7 (+4.7/+0.0)	14.8 (+2.5/+0.3)
+ Ensemble	11.0 (+5.0/+0.3)	15.4 (+3.1/+0.6)

Table 13: Overview of our WMT20 English ↔ Pashto systems. BLEU scores are reported on the offline official validation set

3.7 English ↔ Pashto

Our Pashto systems used the similar process we described in the Japanese tasks. We cleaned the 1 million original parallel dataset and kept 700K pairs. BPE was jointly learned and the merge operations count is 10000, but the source language does not share vocabulary with the target. When training the models, the learning rate was set to 9×10^{-4} and warmup steps was 6000. The overview of our En ↔ Ps system is listed in Table 13

As what we did in the Japanese tasks, we selected 10000 sentence pairs from the training dataset according to the test data using FDA, mixed them with official validation set and devtest set to fine-tune our models for 5 epoch, then reranked the generated candidates. This improved our EnPs system by 1.6 BLEU and for PsEn the gain is 3.5.

3.8 English ↔ Polish

For En ↔ Pl tasks, we generally followed the process depicted in En ↔ De tasks, cleaned 10.3 million data pairs and kept 5.265 million. The overview of our En ↔ Pl system is listed in Table

System	EnPl BLEU	PlEn BLEU
Baseline	24.9 (-/-)	30.0 (-/-)
+ Back-translation	28.2 (+3.3/+3.3)	33.0 (+3.0/+3.0)
+ Knowledge distillation	28.8 (+3.9/+0.6)	34.6 (+4.6/+1.6)
+ Ensemble	29.9 (+5.0/+1.1)	35.1 (+5.1/+0.5)
+ Reranking	30.0 (+5.1/+0.1)	35.5 (+5.5/+0.4)

Table 14: Overview of our WMT20 English ↔ Polish systems. Reranking follows noisy-channel reranking. BLEU scores are reported on official released validation dataset.

14. Training methods listed are generally the same as what we did for En ↔ De, the only difference is we separately trained BPE for the two languages (so obviously they no longer share the vocabulary), but BPE merge operations count is still set to 32K.

3.9 English ↔ Russian

The data preprocessing for En ↔ Ru tasks is the same as demonstrated in the En ↔ Km part, the only difference is for Russian, our BPE merge operations count is set to 36K. The official released parallel dataset (without official synthetic dataset) is reduced from 43.8 million pairs to 26.5 million after the cleaning. For Russian tasks, we trained the model with some extra rounds of back-translation and knowledge distillation, which are:

- In the first round back-translation, we only used all the official released data including the synthetic part. After training had converged, we continued training on the parallel dataset.
- In the second round back-translation, we added in the back-translated results generated by our models, and continued training again.
- In the knowledge distillation step, we added in the knowledge distillation results on the base of the dataset produced in the previous step. After training had converged, models are continue trained using the mixture of original parallel dataset and the knowledge distillation results.

Full results can be referred to Table 15.

3.10 English ↔ Tamil

Similar to Khmer, Tamil language also uses abugida. So with the same idea, we need to determine the minimal unit to be separated during BPE training. Here we see syllables as the minimal unit, use *open-tamil* [4] to separate syllables, and use our modified *subword-nmt* to learn BPE separations. Cleaning process is the same as we described in the Khmer tasks, we cleaned all the parallel corpora which contains 660K pairs, and had 450K pairs left. For back-translation, we used all available Tamil monolingual corpus (27 million lines totally) and 16 million English sentences sampled from NewsCrawl 2019 and NewsCommentary 2019. BPE is learned jointly, the merge operations count is 10K. The overview of our En ↔ Ta system is listed in Table 16. In the fine-tune stage, we randomly kept 200 sentences from the newsdev2020 as the validation set, and the rest 1,789 sentences are used to fine-tune the model.

System	EnRu BLEU	RuEn BLEU
Baseline	32.1	38.7 (-/-)
+ Bigger ffn dim	32.6 (+0.5/+0.5)	38.8 (+0.1/+0.1)
+ 1st. round back-translation	32.7 (+0.6/+0.1)	39.0 (+0.3/+0.2)
+ 2nd. round back-translation	33.6 (+1.5/+0.9)	39.6 (+0.9/+0.6)
+ knowledge distillation	34.1 (+2.0/+0.5)	40.4 (+1.7/+0.8)
+ Fine-tune	35.2 (+3.1/+1.1)	40.9 (+2.2/+0.5)
+ Ensemble	35.7 (+3.6/+0.5)	41.3 (+2.6/+0.4)
+ Reranking	35.5 (+3.4/-0.2)	41.7 (+3.0/+0.4)

Table 15: Overview of our WMT20 English ↔ Russian systems. BLEU scores are reported on newstest2019. "Bigger ffn dim" means we augmented the dimension of fast forward layer to 8192. In the step "Fine-tune" we fine-tuned our models using the mixture of newstest2017 and newstest2018

System	EnTa BLEU	TaEn BLEU
Baseline	7.6 (-/-)	14.4 (-/-)
+ Back-translation	13.1 (+5.5/+5.5)	26.2 (+11.8/+11.8)
+ Fine-tune*	20.2 (+12.6/+7.1)	31.5 (+17.1/+5.3)
+ Ensemble*	20.4 (+12.8/+0.2)	32.5 (+18.1/+1.0)
+ Reranking*	21.6 (+14.0/+1.2)	32.7 (+18.3/+0.2)

Table 16: Overview of our WMT20 English ↔ Tamil systems. BLEU scores are reported on newsdev2020. Configurations can be referred to the Khmer tasks. Steps with extra ∗ marks are evaluated in the tiny 200 lines new validation set.

3.11 French ↔ German

Our Fr↔De systems generally followed the steps we described in the Russian tasks, with two differences. The first is that we have only one round back-translation, since for this task pair no official back-translation dataset was released; the second is we didn't continue training using parallel corpus after the model had converged. Following the process described in the Khmer tasks, we cleaned the 13.7 million data pairs and kept 11 million. For

[4] https://github.com/Ezhil-Language-Foundation/open-tamil

System	FrDe BLEU	DeFr BLEU
Baseline	28.9 (-/-)	35.4 (-/-)
+ Back-translation	36.2 (+7.3/+7.3)	36.4 (+1.0/+1.0)
+ knowledge distillation	36.2 (+7.3/+0.0)	36.6 (+1.2/+0.2)
+ Fine-tune	36.3 (+7.4/+0.1)	37.6 (+2.2/+1.0)
+ Ensemble (4 models)	36.7 (+7.8/+0.4)	37.9 (+2.5/+0.3)
+ Reranking	36.8 (+7.9/+0.1)	38.1 (+2.7/+0.2)

Table 17: Overview of our WMT20 French ↔ German systems. BLEU scores are reported on newstest2019. In the step "Fine-tune" we fine-tuned our models using euelections_dev2019

back-translation, we took 27 million French sentences (combination of NewsCrawl 2017-2019 and News Commentary datasets) and 40 million German sentences (from NewsCrawl 2019 only). We jointly learned BPE for the two langauges, the BPE merge operations count is 32K. We shared the vocabulary among the two languages and tied all embedding layers and output layer in the model. The overview of our Fr ↔ De system is listed in Table 17.

4 Conclusion

This report described OPPO's submissions to the WMT20 news translation task. We use the similar data preprocess and filtering strategy for all the tasks, contains statistical information based rules and alignment information based rules. We trained Transformer-Big models for all the directions and applied some mature techniques, like back-translation, ensemble model, fine-tune and reranking, they generally all brought gains for the final results. Our final submissions ranked top in 6 directions (English ↔ Czech, English ↔ Russian, French → German and Tamil → English), third in 2 directions (English → German, English → Japanese), and fourth in 2 directions (English → Pashto and and English → Tamil).

References

Ergun Biçici and Deniz Yuret. 2011. Instance selection for machine translation using feature decay algorithms. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 272–283.

Franck Burlot and François Yvon. 2018. Using monolingual data in neural machine translation: a systematic study. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 144–155.

Colin Cherry and George Foster. 2012. Batch tuning strategies for statistical machine translation. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 427–436. Association for Computational Linguistics.

Chris Dyer, Victor Chahuneau, and Noah A Smith. 2013. A simple, fast, and effective reparameterization of ibm model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500.

Markus Freitag, Yaser Al-Onaizan, and Baskaran Sankaran. 2017. Ensemble distillation for neural machine translation. *arXiv preprint arXiv:1702.01802*.

Vu Cong Duy Hoang, Philipp Koehn, Gholamreza Haffari, and Trevor Cohn. 2018. Iterative back-translation for neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 18–24.

Chea Sok Huor, Ros Pich Hemy, and Vann Navy. 2004. Detection and correction of homophonous error word for khmer language. *Ref. No. PANL10n/Admn/RR*.

Marcin Junczys-Dowmunt. 2018. Dual conditional cross-entropy filtering of noisy parallel corpora. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 888–895.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia. Association for Computational Linguistics.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Tom Kocmi, Martin Popel, and Ondřej Bojar. 2020. Announcing czeng 2.0 parallel corpus with over 2 gigawords. *arXiv preprint arXiv:2007.03006*.

Ruixuan Luo, Jingjing Xu, Yi Zhang, Xuancheng Ren, and Xu Sun. 2019. Pkuseg: A toolkit for multi-domain chinese word segmentation. *CoRR*, abs/1906.11455.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of*

the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations), pages 48–53.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

Peng Qi, Timothy Dozat, Yuhao Zhang, and Christopher D Manning. 2018. Universal dependency parsing from scratch. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 160–170.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Kyra Yee, Yann Dauphin, and Michael Auli. 2019. Simple and effective noisy channel modeling for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5700–5705.

HW-TSC's Participation in the WMT 2020 News Translation Shared Task

Daimeng Wei[1], Hengchao Shang[1], Zhanglin Wu[1], Zhengzhe Yu[1], Liangyou Li[2], Jiaxin Guo[1], Minghan Wang[1], Hao Yang[1], Lizhi Lei[1], Ying Qin[1], Shiliang Sun[3],
[1]Huawei Translation Service Center, Beijing, China
[2]Huawei Noah's Ark Lab, Hong Kong, China
[3]East China Normal University, Shanghai, China
{weidaimeng,shanghengchao,wuzhanglin2,yuzhengzhe,liliangyou,
guojiaxin1,wangminghan,yanghao30,leilizhi,qinying}@huawei.com
slsun@cs.ecnu.edu.cn

Abstract

This paper presents our work in the WMT 2020 News Translation Shared Task. We participate in 3 language pairs including Zh/En, Km/En, and Ps/En and in both directions under the constrained condition. We use the standard Transformer-Big model as the baseline and obtain the best performance via two variants with larger parameter sizes. We perform detailed pre-processing and filtering on the provided large-scale bilingual and monolingual dataset. Several commonly used strategies are used to train our models such as Back Translation, Ensemble Knowledge Distillation, etc. We also conduct experiment with similar language augmentation, which lead to positive results, although not used in our submission. Our submission obtains competitive results in the final evaluation.

1 Introduction

This paper introduces our work for the WMT 2020 News Translation Shared Task. We participate in three language pairs including Chinese/English (Zh/En), Khmer/English (Km/En), Pashto/English (Ps/En) and in both directions. After observation, we consider that the officially provided dataset has the acceptable size and quality therefore only participate in the constrained evaluation. Our method is mainly based on previous works but with fine-grained data cleaning techniques and language pair specific optimizations.

For each language pair, we perform careful multi-step cleaning on the provided dataset and only keep a high-quality subset for training. At the same time, several strategies are tested in a pipeline including Back-Translation (Edunov et al., 2018), Ensemble Knowledge Distillation (Freitag et al., 2017; Li et al., 2019), Forward Translation (Wu et al., 2019), Fine-Tuning (Sun et al., 2019), and Ensemble and Re-ranking (Ng et al., 2019a).

Due to the page limitation, we mainly introduce our methods and experiments on the Zh-En and En-Zh language pairs. Most of these methods are also employed on the Km/En and Ps/En pairs. Special optimizations regarding different language will be introduced separately.

2 Data

In this section, we describe the size and source of the dataset as well as our cleaning and filtering techniques.

2.1 Data Source

2.1.1 Zh/En

We use both bilingual and monolingual text to train the model. Regarding bilingual text, we merge the data from CCMT (7M), Wiki Titles v1 (1M), News Commentary v15 (0.4M) and a subset of UN Parallel Corpus (9M). We also select 10 million of Zh and En monolingual text from Xin Hua, XMU and News crawl respectively for back translation.

2.1.2 Km/En

We use the Para Crawl v5.1 (4.17M), Khmer and Pashto parallel data (0.29M) as the bitext corpus, and select 10M monolingual text from Common Crawl and news crawl 2018 for Km and En, respectively.

2.1.3 Ps/En

Similar to Km/En, we also use the Para Crawl v5.1 (1M), Khmer and Pashto parallel data (0.03M) as bitext and select 6.5M monolingual text from Common Crawl and news crawl 2018.

2.2 Data Pre-processing

For the Zh/En corpus, we use following operations to pre-process the data:

Proceedings of the 5th Conference on Machine Translation (WMT), pages 293–299

Operation	Zh-En			Km-En			Ps-En		
	Zh-En (bi)	Zh (mono)	En (mono)	Km-En (bi)	Km (mono)	En (mono)	Ps-En (bi)	Ps (mono)	En (mono)
Original	21	21.4	18	4.46	12.59	10	1.05	6.6	4.71
+ Deduplication	20.9	21.3	17.9	4.30	12.57	9.99	1.05	5.99	4.70
+ Lang-id filtering	20.4	19.6	17.9	2.82	11.13	9.93	1.02	5.69	4.38
+ Length filtering	20.1	19	17.9	2.71	10.54	9.90	0.94	4.97	4.14
+ Fast-align filtering	19.5	-	-	0.8	-	-	0.54	-	-
+ Data-selection	16.5	10	10	-	-	-	-	-	-

Table 1: This table shows the remaining data size of performing specific data cleaning and selection operations, where the unit is million (M). The bilingual (bi) and monolingual (mono) texts are both listed in the table for all three language pairs.

- Regarding Chinese text, we tokenize the text with Jieba[1] tokenizer, and create the BPE (Sennrich et al., 2016) vocab with 30K merge operations.
- For English text, we use moses[2] tokenizer and generate a BPE vocab with 32K merge operations.
- Bitexts with length ratios (source/target) greater than 3 are removed.
- Texts longer than 120 sub-tokens are removed.
- Texts with undesired fastText-langid (Joulin et al., 2016b,a) are removed.

For the Km/En and Ps/En corpus, following operations are performed on the data:

- Full-width texts are converted to half-width texts.
- De-duplication is performed.
- Texts which the source or target is empty are empty.
- Sentences with undesired fastText-langid (Joulin et al., 2016b,a) are removed.
- SPM with regularization (Kudo and Richardson, 2018; Kudo, 2018) is used for both language pairs.
- Fast-align (Dyer et al., 2013) is used to further clean the corpus.
- Sentences with more than 100 sub-tokens are removed.

[1]https://github.com/fxsjy/jieba
[2]http://www.statmt.org/moses/

During experiment, we notice that Km and Ps data has relatively low qualities, which need to be further cleaned in a stricter manner. Therefore, we gradually increase the threshold of fast-align, and remove about 50% of un-aligned text to improve the training data quality. Detailed data size of each step is shown in Table 1.

2.3 Data Selection

Data selection filters out bilingual or monolingual out-of-domain text from a given corpora. We perform data selection on the Zh/En UN dataset, of which the domain is different from news. To do so, we train a classifier to select texts classified as news from the UN corpus. In terms of the classifier, when selecting En→Zh bi-text, we sample the target language (Zh) text from UN and non-UN dataset with an equal size (e.g. 50000), and label them with UN and news tags. Then, we train a Fasttext (Bojanowski et al., 2017) classifier on the sampled set, and score the leftover UN set with the classification probability $P(y = \text{news}|x)$ to retrieve the top-k bi-text pairs, where k is set to 9M in the experiment. Note that even if the score is lower than 0.5, we still keep the sample if its rank is within top-k. This method is also used for Zh→En selection. Note that the selected En→Zh and Zh→En set can be overlapped but not exactly the same.

From the experiment, we find that data selection is quite effective in improving the BLEU score on WMT 2019 test set compared to using entire UN set with a 1.1 increase on Zh→En and a 1.6 increase on En→Zh, respectively.

For the Km/En and Ps/En pairs, we do not employ the data selection strategy, but carefully evaluate the performance of different sources in the training set and finally select the Common Crawl (Km) and News Crawl (En) as the monolingual corpus. KenLM (Heafield, 2011) is also used to filter the data.

3 System Overview

This section describes the model and techniques of our work. We basically perform such strategies sequentially. Our experimental result will be presented on each part.

3.1 Model

Transformer (Vaswani et al., 2017) has been widely used for machine translation in recent years, which has achieved good performance even with the most primitive architecture without much modifications. Therefore, we choose to start from Transformer-Big and consider it as a baseline. Two variants of Transformer are also evaluated during the experiments, which are the model with wider FFN layers proposed in (Ng et al., 2019b), and the deeper encoder version proposed in (Sun et al., 2019). Here, we call two variants Transformer-Large and Transformer-Deep. Our models are implemented with THUMT (Zhang et al., 2017), and trained on a platform with 8 V100 GPUs.

3.2 Back Translation

Following (Edunov et al., 2018), we use back translation (BT) to improve the system performance. However, unlike (Edunov et al., 2018), we use beam search to decode the pseudo source text because in the experiment we find that results from beam search is better than sampling.

To acquire better monolingual text, we also use the method introduced in the data selection section to filter the in-domain subset for BT. For Zh→En and En→Zh direction, we use texts in target language from our bilingual corpus as the in-domain set, monolingual corpus as the out-of-domain set to train the classifier, and finally select approximately 10 million of samples for each direction. The back translated corpus are merged with the original corpus, which improves the performance by 0.6 for Zh→En and 1.3 for En→Zh. For the Km/En pair, we use exactly the same method as Zh/En, but with monolingual corpus from specific language, resulting in improvements of 5.33 and 2.55 in terms of BLEU for Km→En and En→Km on the devtest 20. For Ps/En, BT is performed on the selected data described in previous section, achieving improvements of 8.08 (Ps→En) and 2.89 (En→Ps) in terms of BLEU on each direction.

3.3 Ensemble Knowledge Distillation

Ensemble Knowledge Distillation (Freitag et al., 2017; Li et al., 2019) improves the performance of a student model by distilling knowledge from a group of trained teacher model into it. Comparing with some soft label distillation methods, the EKD for NMT is relatively straightforward, which can be implemented by training the student on the combination of the original training set and the translation from the ensembled teacher model on the training set. In our experiments we ensemble four models as the teacher model to translate the training set. Then, compute the BLEU for each sentence against the ground truth target. We keep 2/3 of the top scored translations for distillation and merge them into the original training set.

Generally speaking, EKD can be performed in an iteration manner. However, this could bring negative influence on the final ensemble. Therefore, we only do it once. EKD improves the BLEU by 1.5 points on the Zh→En direction, but only 0.2 points on the En→Zh direction.

We didn't perform the EKD on the Km/En and Ps/En pairs due to the limitation of the corpus size.

3.4 Forward Translation

As described in (Wu et al., 2019), similar to back translation, the monolingual corpus in source language can also be used to create the forward translation text with a trained MT model, and the created forward and backward translation corpus can both be merged with the original bilingual data. This strategy can enlarge the data size to a large extent. There are basically four steps to perform the forward translation. Take En→Zh as an example: 1) train M models with EKD in both direction; 2) create pseudo corpus with the ensemble of M models on the monolingual corpus in both direction (SRC→TGT', TGT→SRC'); 3) merge the created corpus with others (BT + FT + EKD + bilingual). 4) train a new model on the mixed corpus. This technique improves the performance by 1.0 in terms of BLEU on En→Zh direction and 0.4 BLEU on Zh→En direction. We also perform this strategy on Km/En and Ps/En, which achieves the improvements of 2.50 and 1.17 on En→Km and Km→En directions; 0.18 and 0.65 on En→Ps and Ps→En directions.

Note that the model trained with this technique can be ineffective for ensemble, which means such training strategy might decrease the model diver-

sity.

3.5 Fine-tuning

Previous works demonstrate that fine-tuning a model on in-domain data such as last year's test set could effectively improve the performance of this year (Sun et al., 2019). In the experiment, we fine-tune the model on the newstest18 for Zh→En with 3000 tokens per batch for one epoch, successfully achieving 3.6 of BLEU improvements on the newstest19. Furthermore, we keep the test corpus with orilang as Zh from newstest18 for fine-tuning, gaining an additional 1.0 BLEU increase. However, this method only obtains 0.2 BLEU increase on the En→Zh direction.

Km/En and Ps/En are newly introduced language pairs in the evaluation this year, thereby have no previous test sets. Since an additional devtest set is provided in addition to the dev set, we fine-tune models on the dev set and test on the devtest set. The experiment shows that fine-tuning could achieve 5.12 and 0.13 of improvements for En→Km and Km→En; 0.59 and 0.79 for En→Ps and Ps→En.

3.6 Ensemble

Six Transformer models are trained with different seeds, including 2 deep, 2 big and 2 large variants. The ensemble model improves the performance by 1.0 on Zh→En and 0.4 on En→Zh in terms of BLEU.

For Km/En and Ps/En pairs, we trained 4 and 6 Transformer-Deep models for Km/En and Ps/En. However, due to the size limitation, the improvements of ensemble is not significant for these two language pairs.

3.7 Ensemble MT Fine-tuning

We perform an additional experiment, named Ensemble MT Fine-tuning. First of all, we fine-tune 6 models on the 18 test set and produce the translation (mt) with the ensemble of them on the 19 test set. Then, we fine-tune the un-fine-tuned 6 models with the mt, which surprisingly improves about 0.6 BLEU on En→Zh. But we see no improvements on Zh→En. This experiment is also performed on Km/En and Ps/En language pairs, but only obtains limited improvements.

While submission, we fine-tune all 6 models on 18 test set and produce the mt with the ensemble model on the 20 test set. We then use the mt of 20 test set to fine-tune the original un-fine-tuned model to get the final one.

	Zh→En	
System	news2018	news2019
baseline	24.98	25.76
+ Data Selection	25.44	26.89 (+1.1)
+ Back-Translation	27.11	27.49 (+0.6)
+ EKD	27.18	29.06 (+1.5)
+ Forward-Translation	28.55	30.45 (+0.4)
+ Fine-tuning	-	35.07 (+4.6)
+ Ensemble	-	36.11 (+1.0)
+ Ensemble MT Fine-tune	-	36.11 (+0.0)
2020 Submission	34.3	

Table 2: The experimental result of Zh→En

	En→Zh	
System	news2018	news2019
baseline	37.84	34.86
+ Data Selection	38.91	36.47 (+1.6)
+ Back-Translation	44.29	38.48 (+1.3)
+ EKD	44.19	38.68 (+0.2)
+ Forward-Translation	43.79	39.69 (+1.0)
+ Fine-tuning	-	39.89 (+0.2)
+ Ensemble	-	40.32 (+0.4)
+ Ensemble MT Fine-tune	-	41.00 (+0.6)
2020 Submission baseline	46.0	

Table 3: The experimental result of En→Zh

3.8 Re-ranking

We also tested the noisy channel re-ranking proposed in (Ng et al., 2019b). However, we do not see consistent improvements on the news2019 and devtest set, thus we give up using the strategy in the submission for all three language pairs.

3.9 Similar Language Augmentation

We also investigate whether performing data augmentation with corpora in similar languages can boost system performances on low resource tasks like En/Km and En/Ps. Inspired by (Kudugunta et al., 2019) who propose the concept of language similarity that can be measured by the SVCCA score on hidden representations of a language pair.

We select top-two similar languages for Km and Ps, by referring to the (Kudugunta et al., 2019). We then collect a set of bilingual text from these languages and mix them into the original training set. For Ps, we collect bilingual corpus of Persian (Fa) and Urdu (Ur) for augmentation, and create

System	Km→En dev	devtest
baseline	7.54	5.90
+ Strict Fast-align	10.63	8.69 (+2.79)
+ Back-Translation	16.48	14.02 (+5.33)
+ Forward-Translation	18.04	15.19 (+1.17)
+ Fine-tuning	-	15.32 (+0.13)
+ Ensemble	-	15.47 (+0.15)
2020 Submission		25.33

Table 4: The experimental result of Km→En

System	En→Km dev	devtest
baseline	29.27	27.93
+ Strict Fast-align	41.39	37.72 (+9.79)
+ Back-Translation	44.61	40.27 (+2.55)
+ Forward-Translation	46.81	42.77 (+2.50)
+ Fine-tuning	-	47.89 (+5.12)
+ Ensemble	-	48.46 (+0.57)
2020 Submission		58.58

Table 5: The experimental result of En→Km. Note that the BLEU score of the dev and devtest are calculated with sentences tokenized with char-based tokenizer.

System	Ps→En dev	devtest
baseline	5.43	6.9
+ Strict Fast-align	7.4	7.31 (+0.41)
+ Back-Translation	14.96	15.39 (+8.08)
+ Forward-Translation	15.87	16.04 (+0.65)
+ Fine-tuning	-	16.83 (+0.79)
+ Ensemble	-	17.25 (+0.42)
2020 Submission		23.1

Table 6: The experimental result of Ps→En

a mixed corpora (Ps:Fa:Ur=8:10:3) with a size of 2.6M, much larger than the original bitext corpus. For Km, we collect Polish (Pl) and Corsican (Ca) as the augmentation language (Km:Pl:Ca=2:2:1) and mix them with the total size of 2.1M.

The experimental result shows that the augmentation improves the BLEU score by 1-3 points on all directions compared to merely training on the original training set, demonstrating that incorporate data of similar languages for data augmentation is effective. However, this advantage disappears when comparing with the strategy of using the Forward and Backward Translation with original language pair, because BT and FT fill the gap of the difference in the data size, and thereby fills the gap of the performance.

System	En→Ps dev	devtest
baseline	4.15	4.3
+ Strict Fast-align	6.0	6.13 (+1.83)
+ Back-Translation	9.01	9.02 (+2.89)
+ Forward-Translation	9.3	9.2 (+0.18)
+ Fine-tuning	-	11.02 (+0.59)
+ Ensemble	-	11.44 (+0.42)
2020 Submission		12.1

Table 7: The experimental result of En→Ps

Although this strategy works fine on a corpus with limited size, it is not as feasible as BT. At the same time, we understand that applying external similar language corpora is not allowed in the constrained track, and finally give up this method. But we would like to conduct further researches on this direction.

4 Results

This section presents the experimental results for each direction of all three language pairs in Table 2,3,4,5,6and 7, where the contribution of strategies introduced in previous sections are listed in each row.

5 Analysis

Here are several findings worthy of sharing during our experiments:

- We test different combinations of model architectures for ensemble, and find that the heterogeneous combinations often perform better than homogeneous combinations when the performance of each model is similar. We suppose that heterogeneous architectures are good at learning different kinds of patterns, which is potentially effective for ensemble.

- While performing data selection, we also test language models as described in (Ng et al., 2019b), but found that fasttext performed better than LMs. We consider this finding is

relatively intuitive because the objective of training the classifier could naturally distinguish features of inter-class samples and cluster inner-class samples, which should be more efficient than using LMs.

- When we perform back-translation and forward-translation on Km/En pairs, we find that no matter in which direction, monolingual text from news domain performs consistently better than that from wiki domain, but the bilingual texts are actually from wiki. The reason for the performance improvements contributed by news corpus might be that the size of the filtered bilingual corpus is small, therefore requires to learn more semantic patterns from BT and FT. Such semantic patterns appear more often in news corpus and thus surpass the loss caused by domain shifting.

6 Conclusion

This paper presents the submissions by HW-TSC on the WMT 2020 News Translation Task. For each direction in three language pairs, we perform experiments with a series of pre-processing and training strategies. The effectiveness of each strategy is demonstrated. Our experiments on similar language augmentation shows that corpora with similar languages can be used for performance improvements in low resource scenarios. Our submission finally achieves competitive result in the evaluation.

References

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Trans. Assoc. Comput. Linguistics*, 5:135–146.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM model 2. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA*, pages 644–648.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 489–500.

Markus Freitag, Yaser Al-Onaizan, and Baskaran Sankaran. 2017. Ensemble distillation for neural machine translation. *CoRR*, abs/1702.01802.

Kenneth Heafield. 2011. Kenlm: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation, WMT@EMNLP 2011, Edinburgh, Scotland, UK, July 30-31, 2011*, pages 187–197.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016a. Fasttext.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016b. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 66–75.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018: System Demonstrations, Brussels, Belgium, October 31 - November 4, 2018*, pages 66–71.

Sneha Reddy Kudugunta, Ankur Bapna, Isaac Caswell, and Orhan Firat. 2019. Investigating multilingual NMT representations at scale. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 1565–1575.

Bei Li, Yinqiao Li, Chen Xu, Ye Lin, Jiqiang Liu, Hui Liu, Ziyang Wang, Yuhao Zhang, Nuo Xu, Zeyang Wang, Kai Feng, Hexuan Chen, Tengbo Liu, Yanyang Li, Qiang Wang, Tong Xiao, and Jingbo Zhu. 2019. The niutrans machine translation systems for WMT19. In *Proceedings of the Fourth Conference on Machine Translation, WMT 2019, Florence, Italy, August 1-2, 2019 - Volume 2: Shared Task Papers, Day 1*, pages 257–266.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019a. Facebook fair's WMT19 news translation task submission. In *Proceedings of the Fourth Conference on Machine Translation, WMT 2019, Florence, Italy, August 1-2, 2019 - Volume 2: Shared Task Papers, Day 1*, pages 314–319.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019b. Facebook FAIR's WMT19 News Translation Task Submission. *arXiv preprint arXiv:1907.06616*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*.

Meng Sun, Bojian Jiang, Hao Xiong, Zhongjun He, Hua Wu, and Haifeng Wang. 2019. Baidu neural machine translation systems for WMT19. In *Proceedings of the Fourth Conference on Machine Translation, WMT 2019, Florence, Italy, August 1-2, 2019 - Volume 2: Shared Task Papers, Day 1*, pages 374–381.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, \Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Lijun Wu, Yiren Wang, Yingce Xia, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. 2019. Exploiting monolingual data at scale for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 4205–4215.

Jiacheng Zhang, Yanzhuo Ding, Shiqi Shen, Yong Cheng, Maosong Sun, Huan-Bo Luan, and Yang Liu. 2017. THUMT: an open source toolkit for neural machine translation. *CoRR*, abs/1706.06415.

IIE's Neural Machine Translation Systems for WMT20

Xiangpeng Wei[1,2], **Ping Guo**[1,2], **Yunpeng Li**[1,2], **Xingsheng Zhang**[1,2], **Luxi Xing**[1,2], **Yue Hu**[1,2]
[1]Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[2]School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
{weixiangpeng,huyue}@iie.ac.cn

Abstract

In this paper we introduce the systems IIE submitted for the WMT20 shared task on German↔French news translation. Our systems are based on the Transformer architecture with some effective improvements. Multiscale collaborative deep architecture, data selection, back translation, knowledge distillation, domain adaptation, model ensemble and re-ranking are employed and proven effective in our experiments. Our German→French system achieved 35.0 BLEU and ranked the second among all anonymous submissions, and our French→German system achieved 36.6 BLEU and ranked the fourth in all anonymous submissions.

1 Introduction

We participate in the WMT20 shared news translation task in one language pair and two language directions, German→French and French→German. Our methods are based on techniques and approaches used in submissions from past years (Deng et al., 2018; Ng et al., 2019; Sun et al., 2019; Li et al., 2019; Xia et al., 2019), including the use of subword models (Sennrich et al., 2016), iterative back-translation, knowledge distillation, model ensembling and several techniques we proposed recently (Wei et al., 2020b,a).

For our submissions of two language directions, we adopt the deep transformer architectures (48-layer) based on multiscale collaboration mechanism (Wei et al., 2020b) as our baseline, which outperformed the standard `Transformer-Big` as well as shallower models significantly in terms of translation quality. We also use an iterative back-translation approach (Zhang et al., 2018) with the controllable sampling to extend the back translation method by jointly training source-to-target and target-to-source NMT models. Moreover, the knowledge distillation (Freitag et al., 2017) is employed to leverage the source-side monolingual data. For our final models, we apply a domain-specific fine-tuning process and model ensembling, and decode using noisy channel model re-ranking.

The paper is structured as follows: Section 2 describes the techniques we used, then section 3 shows the experimental settings and results. Finally, we conclude our work in Section 4.

2 Our Techniques

2.1 Multiscale Collaborative Deep Models

The structure of NMT models has evolved quickly, such as RNN-based (Wu et al., 2016), CNN-based (Gehring et al., 2017) and attention-based (Vaswani et al., 2017) systems. Deep neural networks have revolutionized the state-of-the-art in various communities, from computer vision to natural language processing. We adopt the deep transformer model proposed by our work (Wei et al., 2020b). Instead of relying on the whole encoder stack to directly learn a desired representation, we let each encoder block learn a fine-grained representation and enhance it by encoding spatial dependencies using a bottom-up network. For coordination, we attend each block of the decoder to both the corresponding representation of the encoder and the contextual representation with spatial dependencies. This not only shortens the path of error propagation, but also helps to prevent the lower level information from being forgotten or diluted. In this section we describe the details (as illustrated in figure 1) of our deep architectures as below:

Block-Scale Collaboration. An intuitive extension of naive stacking of layers is to group few stacked layers into a *block*. We suppose that the encoder and decoder of our model have the same number of blocks (i.e., N). Each block of the encoder has M_n ($n \in \{1, 2, ..., N\}$) identical layers,

Proceedings of the 5th Conference on Machine Translation (WMT), pages 300–304
Online, November 19–20, 2020.

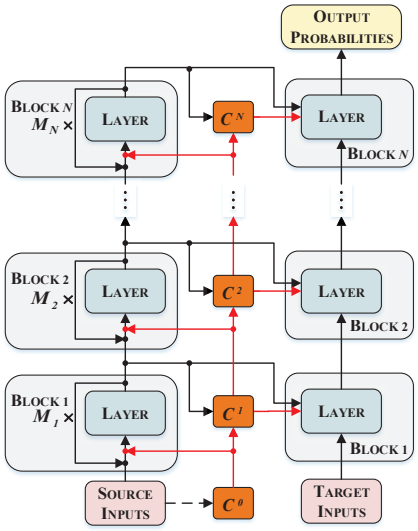


Figure 1: Illustration of Multiscale Collaborative Deep NMT Model. N is the number of encoder and decoder blocks. The n-th block of the encoder consists of M_n layers, while each decoder block only contains one layer.

while each decoder block contains one layer. Thus, we can adjust the value of each M_n flexibly to increase the depth of the encoder. Formally, for the n-th block of the encoder:

$$\mathrm{B}_e^n = \textsc{Block}_e(\mathrm{B}_e^{n-1}), \tag{1}$$

where $\textsc{Block}_e(\cdot)$ is the block function, in which the layer function $\mathcal{F}(\cdot)$ is iterated M_n times, i.e.

$$\begin{aligned} \mathrm{B}_e^n &= \mathrm{H}_e^{n,M_n}, \\ \mathrm{H}_e^{n,l} &= \mathcal{F}(\mathrm{H}_e^{n,l-1}; \mathbf{\Theta}_e^{n,l}) + \mathrm{H}_e^{n,l-1}, \\ \mathrm{H}_e^{n,0} &= \mathrm{B}_e^{n-1}, \end{aligned} \tag{2}$$

where $l \in \{1, 2, ..., M_n\}$, $\mathrm{H}_e^{n,l}$ and $\mathbf{\Theta}_e^{n,l}$ are the representation and parameters of the l-th layer in the n-th block, respectively. The decoder works in a similar way but the layer function $\mathcal{G}(\cdot)$ is iterated only once in each block,

$$\begin{aligned} \mathrm{B}_d^n &= \textsc{Block}_d(\mathrm{B}_d^{n-1}, \mathrm{B}_e^n) \\ &= \mathcal{G}(\mathrm{B}_d^{n-1}, \mathrm{B}_e^n; \mathbf{\Theta}_d^n) + \mathrm{B}_d^{n-1}. \end{aligned} \tag{3}$$

Each block of the decoder attends to the corresponding encoder block.

Contextual Collaboration. To model long-term spatial dependencies and reuse global representations, we define a GRU cell $\mathcal{Q}(\mathbf{c}, \bar{\mathbf{x}})$, which maps a hidden state $\mathbf{c}$ and an additional input $\bar{\mathbf{x}}$ into a new hidden state:

$$\begin{aligned} \mathrm{C}^n &= \mathcal{Q}(\mathrm{C}^{n-1}, \mathrm{B}_e^n), n \in [1, N] \\ \mathrm{C}^0 &= \mathcal{E}_e, \end{aligned} \tag{4}$$

where $\mathcal{E}_e$ is the embedding matrix of the source input $\mathbf{x}$. The new state C^n can be fused with each layer of the subsequent blocks in both the encoder and the decoder. Formally, B_e^n in Eq.(1) can be re-calculated in the following way:

$$\begin{aligned} \mathrm{B}_e^n &= \mathrm{H}_e^{n,M_n}, \\ \mathrm{H}_e^{n,l} &= \mathcal{F}(\mathrm{H}_e^{n,l-1}, \mathrm{C}^{n-1}; \mathbf{\Theta}_e^{n,l}) + \mathrm{H}_e^{n,l-1}, \\ \mathrm{H}_e^{n,0} &= \mathrm{B}_e^{n-1}. \end{aligned} \tag{5}$$

Similarly, for decoder, we have

$$\begin{aligned} \mathrm{B}_d^n &= \textsc{Block}_d(\mathrm{B}_d^{n-1}, \mathrm{B}_e^n) \\ &= \mathcal{G}(\mathrm{B}_d^{n-1}, \mathrm{B}_e^n, \mathrm{C}^n; \mathbf{\Theta}_d^n) + \mathrm{B}_d^{n-1}. \end{aligned} \tag{6}$$

2.2 Back-Translation with Controllable Sampling

Back-translation (BT) is an effective and commonly used data augmentation technique to incorporate monolingual data into a translation system. Back-translation first trains an intermediate target-to-source system that is used to translate monolingual target data into additional synthetic parallel data. This data is used in conjunction with human translated bitext data to train the desired source-to-target system.

In our work, we use an iterative back-translation approach to jointly train source-to-target and target-to-source NMT models. The process can be summarized as below:

- step 1: we train both a source-to-target model ($\mathcal{M}^0_{x\to y}$) and a target-to-source model ($\mathcal{M}^0_{y\to x}$) using the human translated data.
- step 2: we use $\mathcal{M}^t_{x\to y}$ to translate source-side monolingual data to target language, and use $\mathcal{M}^t_{y\to x}$ to translate target-side monolingual data to source language, where t starts from 0.
- step 3: we combine both the human translated data and pseudo data synthesized in step 2 to further optimize the two NMT models respectively.
- Repeat steps 2-3 until the models converge.

In practice, we repeat 3 times for steps 2-3. We apply the controllable sampling strategy (Wei et al., 2020a) to synthesize reasonable sentences which are at both high quality and diversity.

2.3 Knowledge Distillation and Ensemble

The early adoption of knowledge distillation (KD) (Kim and Rush, 2016) is for model compression. We use the same method as in Sun et al. (2019) that adopts hybrid heterogeneous teacher: base transformer, deep transformer, big transformer and RNMT+ (Chen et al., 2018). For each individual model, we use the other two models as the teacher model to further improve the performance. In addition, model ensemble is also used to boost the performance by combining the predictions of above four models at each decoding step.

2.4 Domain-specific Fine-tuning

Fine-tuning with domain-specific data is a common and effective method to improve translation quality for a downstream task. After completing training on the bitext and back-translated data, we train for an additional epoch on a smaller in-domain corpus. We first select 100K sentence-pairs from the bilingual as well as pseudo-generated data according to the filter method in Deng et al. (2018) and continue to train the model on the filtered data.

2.5 Reranking

N-best reranking is a method of improving translation quality by scoring and selecting a candidate hypothesis from a list of n-best hypotheses generated by a source-to-target model. For our submissions, we rerank the n-best hypotheses using two aspects as follows:

$$\log p(y|x) + \lambda_1 \log p(x|y) + \lambda_2 \log p(y) \quad (7)$$

The weights λ_1 and λ_2 are determined by tuning them with a random search on a validation set and selecting the weights that give the best performance.

3 System Overview

We submit constrained systems to both German to French and French to German translations, with the same techniques.

3.1 Dataset

We use all available bilingual datasets and select 10M bilingual data from WMT'20 corpora using the script `filter_interactive.py`[1]. We share a vocabulary for the two languages and apply BPE for word segmentation with 32K merge operations. For monolingual data, we use 18M German sentences and 18M French sentences from Newscrawl, and pre-process them in the same way as bilingual data. We split 9k sentences from the "dev08-14" as the validation set and use newstest 2019 as the test set.

[1]Scripts at: https://tinyurl.com/yx9fpoam.

System	German→French	
	Dev	Newstest19
MSC (48L)	28.9	33.2
+ Iterative BT	31.2	35.7
+ KD & Ensemble	32.3	36.5
+ Fine-tuning	32.9	37.2
+ Reranking	33.8	38.4
WMT'20 submission		**35.0**

Table 1: SacreBLEU scores on German→French.

System	French→German	
	Dev	Newstest19
MSC (48L)	22.8	31.7
+ Iterative BT	24.2	34.0
+ KD & Ensemble	25.1	34.7
+ Fine-tuning	25.9	35.4
+ Reranking	26.5	36.3
WMT'20 submission		**36.6**

Table 2: SacreBLEU scores on French→German.

3.2 Model Configuration

We use the PyTorch implementation of Transformer[2]. We choose the `Transformer_base` setting, in which the encoder and decoder are of 48 and 6 layers, respectively. The dropout rate is fixed as 0.1. We set the batch size as 4096 and the parameter `--update-freq` as 16.

3.3 Results

Results and ablations for De→Fr Fr→De are shown in Table 1 and 2, respectively. We report case-sensitive SacreBLEU scores using SacreBLEU (Post, 2018)[3], using international tokenization for German↔French.

German→French For De→Fr, iterative BT improves our baseline performance on newstest 2019

[2]https://github.com/pytorch/fairseq

[3]SacreBLEU signatures: BLEU+case.mixed+lang.de-fr+numrefs.1+smooth.exp+test.wmt19+tok.13a+version.1.2.11, BLEU+case.mixed+lang.fr-de+numrefs.1+smooth.exp+test.wmt19+tok.13a+version.1.2.11

by about 2.5 BLEU. The addition of KD and model ensemble improves single model performance by 0.8 BLEU, but combining this with fine-tuning and reranking gives us a total of 2 BLEU. Our final submission for WMT20 achieves 35.0 BLEU points for German→French translation (ranked in the second place).

French→German For Fr→De, we see similar improvements with iterative BT by about 2.3 BLEU. KD, ensembling, and fine-tuning add an additional 1.4 BLEU, with reranking contributing 0.9 BLEU. Our final submission for WMT20 achieves 36.6 BLEU points for French→German translation (ranked in the fourth among anonymous submissions).

4 Conclusion

This paper describes CAS IIE's submission to the WMT20 German↔French news translation task. We investigate extremely deep models (with 48 layers) and exploit effective strategies to better utilize parallel data as well as monolingual data. Finally, our German→French system achieved 35.0 BLEU and ranked the second among all anonymous submissions, and our French→German system achieved 36.6 BLEU and ranked the fourth in all anonymous submissions.

References

Mia Xu Chen, Orhan Firat, Ankur Bapna, Melvin Johnson, Wolfgang Macherey, George Foster, Llion Jones, Mike Schuster, Noam Shazeer, Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Zhifeng Chen, Yonghui Wu, and Macduff Hughes. 2018. The best of both worlds: Combining recent advances in neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 76–86, Melbourne, Australia. Association for Computational Linguistics.

Yongchao Deng, Shanbo Cheng, Jun Lu, Kai Song, Jingang Wang, Shenglan Wu, Liang Yao, Guchun Zhang, Haibo Zhang, Pei Zhang, Changfeng Zhu, and Boxing Chen. 2018. Alibaba's neural machine translation systems for wmt18. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 372–380, Belgium, Brussels. Association for Computational Linguistics.

Markus Freitag, Yaser Al-Onaizan, and Baskaran Sankaran. 2017. Ensemble distillation for neural machine translation. *CoRR*, abs/1702.01802.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. In *arXiv:1705.03122*.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327, Austin, Texas. Association for Computational Linguistics.

Bei Li, Yinqiao Li, Chen Xu, Ye Lin, Jiqiang Liu, Hui Liu, Ziyang Wang, Yuhao Zhang, Nuo Xu, Zeyang Wang, Kai Feng, Hexuan Chen, Tengbo Liu, Yanyang Li, Qiang Wang, Tong Xiao, and Jingbo Zhu. 2019. The niutrans machine translation systems for wmt19. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 257–266, Florence, Italy. Association for Computational Linguistics.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook fair's wmt19 news translation task submission. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 314–319, Florence, Italy. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Meng Sun, Bojian Jiang, Hao Xiong, Zhongjun He, Hua Wu, and Haifeng Wang. 2019. Baidu neural machine translation systems for wmt19. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 374–381, Florence, Italy. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 5998–6008.

Xiangpeng Wei, Heng Yu, Yue Hu, Rongxiang Weng, Luxi Xing, and Weihua Luo. 2020a. Uncertainty-aware semantic augmentation for neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020*. Association for Computational Linguistics.

Xiangpeng Wei, Heng Yu, Yue Hu, Yue Zhang, Rongxiang Weng, and Weihua Luo. 2020b. Multiscale collaborative deep models for neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 414–426, Online. Association for Computational Linguistics.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, and Klaus Macherey. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. In *arXiv:1609.08144*.

Yingce Xia, Xu Tan, Fei Tian, Fei Gao, Di He, Weicong Chen, Yang Fan, Linyuan Gong, Yichong Leng, Renqian Luo, Yiren Wang, Lijun Wu, Jinhua Zhu, Tao Qin, and Tie-Yan Liu. 2019. Microsoft research asia's systems for wmt19. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 424–433, Florence, Italy. Association for Computational Linguistics.

Zhirui Zhang, Shujie Liu, Mu Li, Ming Zhou, and Enhong Chen. 2018. Joint training for neural machine translation models with monolingual data. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 555–562. AAAI Press.

The Volctrans Machine Translation System for WMT20

Liwei Wu[1], **Xiao Pan**[1], **Zehui Lin**[2]*, **Yaoming Zhu**[3]†, **Mingxuan Wang**[1], **Lei Li**[1]
[1]ByteDance AI Lab, Beijing, China
{*wuliwei.000, panxiao.94, wangmingxuan.89, lilei.lab*}*@bytedance.com*
[2]Fudan University, Shanghai China
linzh18@fudan.edu.cn
[3]Shanghai Jiao Tong University, Shanghai China
ymZhu@apex.sjtu.edu.cn

Abstract

This paper describes our VolcTrans system on WMT20 shared news translation task. We participated in 14 translation directions. Our basic systems are based on Transformer (Vaswani et al., 2017), with several variants (wider or deeper Transformers, dynamic convolutions). The final system includes text pre-process, data selection, synthetic data generation, advanced model ensemble, and multilingual pre-training.

1 Introduction

We participated in the WMT2020 shared news translation task in 14 directions: English↔Chinese, English↔German, French↔German, English↔Polish, English↔Tamil,English↔Pashto,English↔Khmer, covering language pairs from high to low resources. In this year's translation task, we mainly focus on exploiting self-supervised and unsupervised methods for NMT to make full use of the monolingual data (Lin et al., 2020; Yang et al., 2019).

We aims at building a general training framework which can be well applied to different translation directions. Our models are mainly based on the Transformer (Vaswani et al., 2017). Techniques used in the submitted systems include iterative back-translation, knowledge distillation. We also employed several tricks to improve in-domain BLEU scores, typically in-domain transfer learning. We also experimented with a multilingual pre-training technique which we proposed recently (Lin et al., 2020).

2 Baseline Models

We apply two different NMT skeletons for the shared news translation as our baseline systems. We use the implementations in Fairseq(Ott et al., 2019). All models are trained with Adam optimizer (Kingma and Ba, 2014). We use the "inverse sqrt lr" scheduler with 4000 warm-up steps and set the max learning rate to 5e-4. The betas are (0.9, 0.98). During training, the batches are made of similar length sequences, so we avoid extreme cases where most sequences in the batch are short and we are required to add lots of pad tokens to each of them because one sequence of the same batch is very long. We limit the batch size to 8192 tokens per GPU, to avoid running out of GPU memory. Meanwhile, to achieve a larger batch size to improve the performance(Ott et al., 2018), we set the parameter "update frequency" to 8, and train the model on 8 GPUs, resulting in an actual batch token size = $8192 \times 8 \times 8$. During training, we employ label smoothing of 0.1 and set dropout rate (Hinton et al., 2012) to 0.2.

2.1 Transformer

Following Sun et al. (2019); Wang et al. (2018), we use different architectures for Transformer(Vaswani et al., 2017) to increase the model diversity and potentially get a better ensemble model.

- Transformer 15e6d: According to Sun et al. (2019), a transformer with larger encoder layer number can learn better representation of source sentence and get better BLEU scores. We increase the number of encoder layers from 6 to 15 layers in the transformer big architecture which is the same as the Deeper Transformer in Sun et al. (2019).

- Transformer Mid 25e6d and Transformer Mid 50e6d: To get much better BLEU scores, we further increase the encoder layer number from 6 to 25 (Transformer Mid 25e6d) and 50 (Transformer Mid 50e6d) for the transformer

*Intern at ByteDance
†Intern at ByteDance

Proceedings of the 5th Conference on Machine Translation (WMT), pages 305–312
Online, November 19–20, 2020.

big architecture. However, the model is too large and can not be trained with GPU, so we decrease the feed forward size from 4096 to 3072 and the embedding size from 1024 to 768.

- Transformer 15000ffn. According to Sun et al. (2019), the performance of the Transformer model is largely dependent on the dimensions of feed forward network. We use the same architecture as Bigger Transformer in Sun et al. (2019) which increases the feed forward size from 4096 to 15000, the attention dropout from 0.1 to 0.3 and the relu dropout from 0.1 to 0.3. The number of encoder and decoder layers remains 6.

- Transformer 128hdim and Transformer 256hdim. Bhojanapalli et al. (2020) shows that a transformer model with larger attention dimensions can also get better BLEU score. We increase the head dimension from 64 to 128 (Transformer 128hdim) and 256 (Transformer 256hdim). The number of encoder and decoder layers remains 6.

- DLCL 25layers. Li et al. (2019) proposes a transformer variant call DLCL and shows that this architecture can make deep transformer get higher BLEU.

2.2 Dynamic Convolution

We also apply dynamic convolution (Wu et al., 2019) architectures.

- Dynamic Convolution 7e6d: The dynamic convolution model with 7 encoder layers and 6 decoder layers which is the same architecture proposed in Wu et al. (2019).

- Dynamic Convolution 25e6d: We increase the encoder layer number from 6 to 25. For layers above 7, we set the kernel size to 31.

3 Experiment Techniques

3.1 Parallel Data Up-sampling

According to the experiments, data diversity matters for the whole system. Apart from splitting the monolingual data into several disjoint parts, we sampled the parallel data so that each model has different deviations on the parallel data. We tested bagging sampling (sample with replacement) and up-sampling(sample with replacement under the premise of using all data), experimental results show that when the amount of parallel data is inadequate with respect to the amount of model parameters (such as French↔German, English↔Polish, etc.), the bagging sampling method reduces the performance of the model; while when the amount of parallel data is abundant(such as English ↔ German), the bagging sampling method has no significant effects on the performance. On the contrary, the data up-sampling method never degrades the performance of the model.

mRASP: Multilingual Pre-training We employed a pre-training method mRASP, which pre-trains a universal multilingual neural machine translation model and fine-tune it on specific language directions. Basically, we pre-train a model using the provided parallel data on WMT2020 of English↔Khmer, English↔Inuktitut, French↔German, English↔Polish, English↔Pashto, English↔Tamil, on a shared vocabulary learned from the above parallel data plus provided monolingual data of all related languages. We learn a BPE sub-word vocabulary with 6000 merge operations. We up-sample the data from lower resource language data to balance data amount and only keep tokens that occur more than 10 times. Finally, we obtain a joint vocabulary of about 28000 tokens.

We fine-tuned the pre-trained model, for low-resource directions: Pashto→English, English↔Khmer and English↔Tamil. The baseline model initialized by this method performs better than the randomly initialized baseline model by a large margin. We pre-trained three mRASP models using the same training data: Transformer big, Transformer 15000ffn and Dynamic Convolution. We report in Table 1 the best score in each setting and direction, and find that mRASP significantly outperforms the baseline.

3.2 Tag Back-Translation

Recently, back-translation (Edunov et al., 2018) is a standard method to improve the translation quality by leveraging the large scale monolingual data. Starting from WMT19, the source of the test set is the natural text and the of the test set is the translationese text. We find the tag back-translation (Caswell et al., 2019) method can achieve better BLEU compared with previous methods proposed in Edunov et al. (2018).

Testset	Ps→En	En→Km	Km→En	En→Ta	Ta→En
Random	10.2	39.3	12.7	7.4	14.0
w/ mRASP	13.8	42.8	14.4	9.2	17.9

Table 1: Comparison between randomly initialized baseline model and model initialized from mRASP model

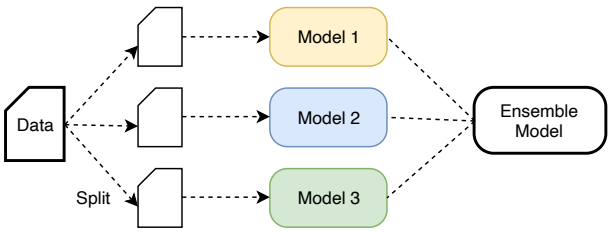


Figure 1: Data Diversity Matters for Final System

Direction	En→Pl	Pl→En
Testset	news20 dev	news20 dev
Baseline	24.8	29.7
Iter 1	27.5	32.6
Iter 2	27.8	32.7
Iter 3	28.2	33.3

Table 2: Iterative Joint Training for English↔Polish

To improve the data diversity among single models before model ensemble, we generated the back-translated data from different monolingual data using different baseline models, as illustrated in Figure 1. For high resource data (English, Polish, etc.), we divided monolingual data into several parts, each containing 10M sentences. However, for low resource data (Pashto), due to the lack of monolingual data, we use all monolingual data for all back-translation tasks.

3.3 Iterative Joint Training

Zhang et al. (2018) proposed an iterative joint training method for better usage of monolingual data from source side and target side. In each iteration, the S2T(source to target) model generates a S2T(target to source) synthetic data from the source side monolingual data and the T2S model generates a T2S synthetic data from the target side monolingual data. Then, the S2T and T2S model are trained with the new T2S and S2T synthetic data to improve the both models performance. In the next iteration, the S2T and T2S model can generate synthetic data with better quality and their performance can be improved further. We jointly trained the S2T and T2S model until they converge. Experiment results on English↔Polish shown in Table 2

3.4 Knowledge Distillation

Recently, knowledge distillation has been widely used to improve the performance of models (Sun et al., 2019; Li et al., 2019). In our knowledge distillation method, student model is trained to fit the output of teacher models. Concretely, we translate the source side monolingual data with an ensemble teacher and a right-to-left(R2L) (Liu et al., 2016) model teacher.

- Ensemble Model. We divided single models in the last joint training iteration into k groups (k=3 in our experiments, resulting in 3 models in each group) and ensemble models in one group to as the teacher model.

- R2L Model. We trained one R2L model for each ensemble group using the same data as anyone model in this group from the last iteration.

We then use pseudo parallel data from ensemble model as well as from R2L model to train the student model, without employing parallel data.

3.5 Advanced Tricks

Top-k Checkpoint Average Different from the conventional checkpoint average approach, which is to average continuous K checkpoints, we average K checkpoints which have the highest BLEU scores on the valid set, and find that this strategy usually leads to significant BLEU improvements over single checkpoints.

Random Ensemble We adopt a simple yet effective strategy in model ensemble. Rather than select the best checkpoint from each model (a.k.a. greedy search), we enlarge the search space: choose one checkpoint from top-k checkpoints from each model, and randomly select N combinations from the entire search space, see Figure 2.

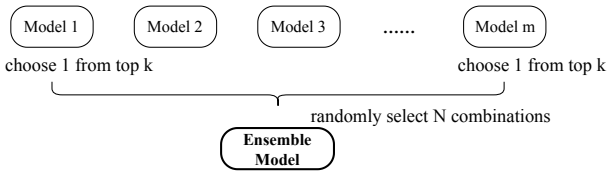


Figure 2: Illustration of Random Ensemble

In domain Fine-tuning There exists a domain mismatch between the obtained system trained with provided parallel or monolingual training data and the target test set. In order to alleviate this mismatch, and to improve the translation performance in the domain of the target test set, we fine-tune the best single models with development sets for 1-2 epochs.

4 Settings and Results

For all news tasks, all ParaCrawl corpus is cleaned by the script proposed by Xia et al. (2019). We trained the baseline using the sampling method described in Section 3.1 with different architectures showed in Section 2. For the low resource language pair (English↔Pashto), as showed in Section 3.1, we pretrained three multilingual models with different model architectures (DLCL 25layers, Transformer 15000ffn and Dynamic Convolution 25e6d) on all parallel data available in WMT20 except the English↔Chinese to avoid a large dictionary[1] and fine-tuned the pre-trained models on the their own parallel data with different data sampling strategies to get 9 baseline models[2]. Then we applied the tag back-translation, joint training, knowledge distillation and random ensemble methods as described in Section 3 to get the final translation system. All BLEU scores were reported with SacreBLEU(Post, 2018).

4.1 Chinese→English

Final Submission We submitted our VolcTrans online system (unconstrained). The final submission achieves 36.6 BLEU. You can get access to VolcTrans online system on `http://translate.volcengine.cn/`.

[1]Large dictionary leads to large parameter size in embedding

[2]We randomly combine sampling strategies and model architectures to get 9 baseline models for each direction. The performances of the 9 baselines is not the point, what we want is the model diversity among single models

Direction	En→Zh
Testset	wmt19
Baseline	38.5
iterative BT	38.9
Ensemble KD	41.5
Ensemble System	42.0
BLEU on WMT20 testset submission	44.9

Table 3: Results of English→Chinese by sacreBLEU

4.2 English→Chinese

For English→Chinese, we train English↔Chinese jointly. We use all parallel data available: News Commentary v15, Wiki Titles v2, UN Parallel Corpus V1.0, CCMT Corpus and WikiMatrix. After data filtering, XM parallel data remained. We use MosesTokenizer for English and Jieba for Chinese. After the pre-processing, separate BPE vocabulary is learned with 32000 merge operations for both English and Chinese on the parallel data. We sample parallel data of ratio 100%, 110% and 120% with replacement from all parallel data. Then we train 3 baselines with Transformer Mid 25e6d, Transformer Mid 50e6d and Dynamic Convolution 25e6d architectures respectively, resulting in 9 baseline models. We employ Newscrawl data as monolingual data for English. The total amount of monolingual data is 90M, containing all Newscrawl 2019 data and others sampled from Newscrawl 2014 to 2018. For Chinese, we employ Newscrawl data, CCMT data and LDC data. We split the Chinese into 3 parts, each contains 8M sentences. For iterative back translation stage and ensemble knowledge distillation stage, each model is combined with different English monolingual data part. Since there are only 3 parts of Chinese monolingual data, we use each part for 3 times at each stage. At the ensemble knowledge distillation stage, we also employ disjoint monolingual data as the distilling data. The detailed results of our system is reported in Table 3.

Final Submission We submitted the ensemble system of the 9 single models after ensemble knowledge distillation stage. The final submission achieves 44.9 BLEU.

4.3 English↔German

For English↔German, we train both directions jointly. We use all parallel data available: Eu-

roparl v10, ParaCrawl v5.1, Common Crawl corpus, News Commentary v15, Wiki Titles v2, Tilde Rapid corpus and WikiMatrix corpus. After data filtering, 28M parallel data remained. We use MosesTokenizer for both English and German. After the pre-processing, a joint BPE vocabulary is learned with 6000 merge operations on the parallel data. We sample parallel data of ratio 80%, 90% and 100% with replacement from all parallel data. Then we train 3 baselines with Transformer Mid 25e6d, Transformer Mid 50e6d and Dynamic Convolution 25e6d architectures respectively, resulting in 9 baseline models. We only employ Newscrawl data as monolingual data for both German and English. The total amount of monolingual data is 90M, containing all Newscrawl 2019 data and others sampled from Newscrawl 2014 to 2018. The 90M data was divided into 9 disjoint parts, each containing 10M sentences, to jointly train 9 systems separately. At the ensemble knowledge distillation stage, we also employ disjoint monolingual data as the distilling data. The detailed results of our system is reported in Table 4.

Direction	En→De		De→En	
Testset	news18	news19	news18	news19
Baseline	47.1	42.2	45.7	41.6
iterative BT	48.6	42.6	48.1	42.2
KD	49.7	44.3	48.4	43.3
Ensemble System	**52.2**	**46.1**	**49.1**	**43.8**
BLEU on WMT20 testset submission	38.2		43.5	

Table 4: Results of English↔German by sacreBLEU

Fine-tune In this step we use the development sets to handle the domain mismatch problem in WMT. For English→German direction, we fine-tune some of the best single models on news2018 for 1-2 epochs, and then get the final ensemble model from models with fine-tune and models without fine-tune.

Final Submission For either direction, the final submission is an ensemble system from single models with highest BLEU scores on development sets. For the final English→German submission, we replaced the English quote with the German quote. The final submissions on Test20 data achieve 38.2 BLEU for English→German direction and 43.5 BLEU for German→English direction.

4.4 French↔German

For French↔German, we train both directions jointly. The overall parallel data contains 13M sentences available including: Europarl v10, ParaCrawl v5.1, Common Crawl corpus, News Commentary v15, Wiki Titles v2 and WikiMatrix corpus. We train 9 baseline models, each with different architectures (Transformer 15e6d * 2, Transformer Mid 25e6d, Transformer Mid 50e6d, Transformer 15000ffn, Transformer 128hdim, Transformer 256hdim and Dynamic Convolution 25e6d) and each group of three models is combined with 3 different sampling strategies (no sample, up sample 120%, up sample 140%)[3], resulting in 9 single models for each direction. We only employ Newscrawl data as monolingual data for both German and French. The monolingual data contains 90M sentences, including all Newscrawl 2019 data and others are sampled from Newscrawl 2014 to 2018. The data of 90M pairs is divided into 9 disjoint parts, each containing 10M sentences to jointly train 9 systems separately. The detailed experiment results are shown in Table 5.

Final Submission For either direction, the final submission is an ensemble system of all 9 models obtained after the knowledge distillation stage. For the final German→French submission, we replaced the English quote with the French quote. The final submissions on Test20 data achieve 35.7 BLEU for French→German direction and 35.3 BLEU for German→French direction.

Direction	Fr→De	De→Fr
Testset	news19	news19
Baseline	26.7	31.3
iterative BT	32.4	35.6
KD	32.7	36.8
Ensemble System	**33.9**	**38.0**
BLEU on WMT20 testset submission	35.7	35.3

Table 5: Results of French↔German by sacreBLEU

4.5 English↔Polish

Our English↔Polish systems are based on Europarl v10, ParaCrawl v5.1, Wiki Titles v2, Tilde Rapid corpus and WikiMatrix corpus. All data add

[3]There is little difference among the models with different sampling ratios, what we are concerned about is the data diversity caused by different sampling ratios.

Direction	En→Pl	Pl→En
Testset	news20 dev	news20 dev
Baseline	24.8	29.7
iterative BT	27.8	32.7
KD	28.2	33.3
Ensemble System	**28.7**	**34.0**
BLEU on WMT20 testset submission	26.1	34.4

Table 6: Results of English↔Polish by sacreBLEU

Direction	En→Ps	Ps→En
Testset	news20 dev	news20 dev
Baseline	8.4	10.2
+mRASP	-	13.8
iterative BT	**9.6**	16.4
Ensemble System	-	**18.0**
BLEU on WMT20 testset submission	10.6	20.0

Table 7: Results of English↔Pashto by sacreBLEU

up to 8M sentences. We train 9 models with different architectures (Transformer 15e6d * 2, Transformer Mid 25e6d, Transformer Mid 50e6d, Transformer 15000ffn, Transformer 128hdim, Transformer 256hdim and Dynamic Convolution 25e6d) and each group of three models is combined with different sampling strategies (no sample, up sample 120%, up sample 140%) on both directions. We only used Newscrawl as English monolingual data. The English monolingual data contains 90M sentences, including all Newscrawl 2019 data and others sampled from Newscrawl 2014 to 2018. We divide the data into 9 disjoint parts. Since Polish Newscrawl corpus only contains 3M sentences, we additionally employ the Polish common crawl data. We sample 90M sentences from the Polish common crawl data which is then divided into 9 disjoint parts. Each part contains 10M common crawl sentences and 3M Newscrawl sentences. Then we apply the joint training and knowledge distillation as described in Section 3. The detailed experiment results are shown in Table 6.

Final Submission Our final submission is an ensemble system consisting of all 9 models obtained after the ensemble knowledge distillation stage. The final submissions on Test20 data achieve 26.1 BLEU for English→Polish direction and 34.4 BLEU for Polish→English direction.

4.6 English↔Pashto

For English↔Pashto, we used all parallel data containing 13M sentences available as follows: ParaCrawl v5.1, Wiki Titles v2 and the Khmer and Pashto parallel data. For Pashto→English, we fine tune the three pre-trained models on all data with different sampling strategies. Each pre-trained model is fine tuned with three different sampling strategies and we get 9 models. For English→Pashto, we find that the models fine-tuned from the pre-trained models have lower BLEU score than the baseline model trained from scratch, so we use the 9 baseline models which are trained with different architectures and sampling strategies. The English monolingual data has 90M sentences containing all Newscrawl 2019 data and others sampled from Newscrawl 2014 to 2018. We divide the data into 9 groups. The detailed experiment results are shown in Table 7.

Final Submission For Pashto→English, our final submission is an ensemble model consisting of all 9 models obtained after the ensemble knowledge distillation stage. For English→Pashto, we find the ensemble model has lower BLEU score than the best single model, so we use the best single model as our final submission. The final submissions achieve 10.6 BLEU on wmt20 testset for English→Pashto direction and 20.0 BLEU for Pashto→English direction.

4.7 English↔Tamil

Direction	En→Ta	Ta→En
Testset	news20 dev	news20 dev
Baseline	7.4	14.0
+mRASP	9.2	17.9
iterative BT	11.8	23.8
BLEU on WMT20 testset submission	7.9	19.7

Table 8: Results of English↔Tamil by sacreBLEU

For English↔Tamil, we use all parallel data containing 533K sentences in total. We use all provided Tamil monolingual data. Following the procedure of English↔German, the English monolingual data contains 90M sentences, including all Newscrawl 2019 data and others sampled from Newscrawl 2014 to 2018. The 90M data was di-

vided into 9 disjoint parts, each containing 10M sentences. For Tamil, we don't apply tokenizer and the raw text is directly split by BPE subword. We fine-tune the three pre-trained models on all parallel data with two different sampling strategies: no sample and up sample 120%, resulting in 6 models. We then conduct back translation for one iteration afterwards, each using one part of the English monolingual data for generating English→Tamil pseudo data, and all Tamil monolingual data for generating Tamil→English pseudo data. The detailed experiment results are shown in Table 8.

Final Submission For both English→Tamil and Tamil→English directions, our final submission is a single model. The final submissions achieve 7.9 BLEU for English→Tamil and 19.7 BLEU for Tamil→English.

4.8 English↔Khmer

Direction	En→Km	Km→En
Testset	news20 dev	news20 dev
Baseline	39.3	12.7
+mRASP	42.8	14.4
iterative BT	46.5	16.9
Ensemble System	-	17.8
BLEU on WMT20 testset submission	51.8	17.6

Table 9: Results of English↔Khmer by sacreBLEU

For English↔Khmer, we used all parallel data containing 4M sentences available as follows: ParaCrawl v5.1 and the Khmer and Pashto parallel data. For Khmer, we extract a dictionary from the Khmer and Pashto parallel data. The km data in this dataset is separated by a special token ǂ00b. Loading this dictionary in the Jieba tokenizer, we get a Khmer tokenizer. We preprocess the Khmer data with our Khmer tokenizer followed by BPE subword. We fine tune the three pre-trained models on all data with different sampling strategies. Each pre-trained model is fine tuned with three different sampling strategies and we get 9 models. We use all provided Khmer monolingual data. The English monolingual data has 90M sentences containing all Newscrawl 2019 data and others sampled from Newscrawl 2014 to 2018. We divide the data into 9 groups. The detailed experiment results are shown in Table 9.

Final Submission For Khmer→English, our final submission is an ensemble model consisting of all 9 models after the iterative back-transkation stage. For English→Khmer, we find the ensemble model has lower BLEU score than the best single model, so we use the best single model as our final submission. The final submissions achieve 51.8 BLEU on wmt20 testset for English→Khmer direction and 17.6 BLEU for Khmer→English direction.

5 Conclusion

This paper describes VolcTrans's NMT systems for the WMT20 shared news translation task. For all directions, we almost adopted the same strategies, except for low-resource language pairs, we employed multilingual pre-training to boost the baseline models. We found that splitting the monolingual data into disjoint parts is an effective way to increase data diversity among single models, which is an important premise for building strong ensemble models. Our final systems achieved significant improvements, usually 3 to 5 BLEU scores, over baseline systems by integrating techniques such as tagged back-translation, iterative back-translation, random ensemble, knowledge distillation.

Acknowledgments

We thank Jun Cao, Zhuo Zhi, Runxin Xu for their support for filtering data, and Zherui Liu for supporting the computing resources. We would also like to thank the anonymous reviewers for their valuable comments.

References

Srinadh Bhojanapalli, Chulhee Yun, Ankit Singh Rawat, Sashank J Reddi, and Sanjiv Kumar. 2020. Low-rank bottleneck in multi-head attention models. *arXiv preprint arXiv:2002.07028*.

Isaac Caswell, Ciprian Chelba, and David Grangier. 2019. Tagged back-translation. *arXiv preprint arXiv:1906.06442*.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. pages 489–500.

Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Bei Li, Yinqiao Li, Chen Xu, Ye Lin, Jiqiang Liu, Hui Liu, Ziyang Wang, Yuhao Zhang, Nuo Xu, Zeyang Wang, Kai Feng, Hexuan Chen, Tengbo Liu, Yanyang Li, Qiang Wang, Tong Xiao, and Jingbo Zhu. 2019. The niutrans machine translation systems for WMT19. In *Proceedings of the Fourth Conference on Machine Translation, WMT 2019, Florence, Italy, August 1-2, 2019 - Volume 2: Shared Task Papers, Day 1*, pages 257–266. Association for Computational Linguistics.

Zehui Lin, Xiao Pan, Mingxuan Wang, Xipeng Qiu, Jiangtao Feng, Hao Zhou, and Lei Li. 2020. Pre-training multilingual neural machine translation by leveraging alignment information. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2649–2663.

Lemao Liu, Masao Utiyama, Andrew M. Finch, and Eiichiro Sumita. 2016. Agreement on target-bidirectional neural machine translation. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 411–416. The Association for Computational Linguistics.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. *arXiv preprint arXiv:1904.01038*.

Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. 2018. Scaling neural machine translation. *CoRR*, abs/1806.00187.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

Meng Sun, Bojian Jiang, Hao Xiong, Zhongjun He, Hua Wu, and Haifeng Wang. 2019. Baidu neural machine translation systems for wmt19. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 374–381.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Mingxuan Wang, Li Gong, Wenhuan Zhu, Jun Xie, and Chao Bian. 2018. Tencent neural machine translation systems for wmt18. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 522–527.

Felix Wu, Angela Fan, Alexei Baevski, Yann N Dauphin, and Michael Auli. 2019. Pay less attention with lightweight and dynamic convolutions. *arXiv: Computation and Language*.

Yingce Xia, Xu Tan, Fei Tian, Fei Gao, Weicong Chen, Yang Fan, Linyuan Gong, Yichong Leng, Renqian Luo, Yiren Wang, et al. 2019. Microsoft research asia's systems for wmt19. *arXiv: Computation and Language*.

Jiacheng Yang, Mingxuan Wang, Hao Zhou, Chengqi Zhao, Yong Yu, Weinan Zhang, and Lei Li. 2019. Towards making the most of bert in neural machine translation. *arXiv preprint arXiv:1908.05672*.

Zhirui Zhang, Shujie Liu, Mu Li, Ming Zhou, and Enhong Chen. 2018. Joint training for neural machine translation models with monolingual data. pages 555–562.

Tencent Neural Machine Translation Systems for the WMT20 News Translation Task

Shuangzhi Wu* SPPD of Tencent
Xing Wang* Tencent AI Lab
Longyue Wang* Tencent AI Lab
Fangxu Liu* SPPD of Tencent
Jun Xie SPPD of Tencent
Zhaopeng Tu Tencent AI Lab
Shuming Shi Tencent AI Lab
Mu Li SPPD of Tencent

Abstract

This paper describes Tencent Neural Machine Translation systems for the WMT 2020 news translation tasks. We participate in the shared news translation task on English ↔ Chinese and English → German language pairs. Our systems are built on deep Transformer and several data augmentation methods. We propose a boosted in-domain finetuning method to improve single models. Ensemble is used to combine single models and we propose an iterative transductive ensemble method which can further improve the translation performance based on the ensemble results. We achieve a BLEU score of 36.8 and the highest chrF score of 0.648 on Chinese → English task.

1 Introduction

Recently, Transformer (Vaswani et al., 2017), that depends on self-attention mechanism , has significantly improved the translation quality. It is widely used as basic Neural Machine Translation (NMT) models in previous WMT translation tasks (Wang et al., 2018b; Li et al., 2019; Sun et al., 2019). In this year's translation task, our Tencent Translation team participated in three WMT2020 shared news translation tasks, including Chinese → English, English → Chinese and English → German. For the three tasks, we use similar model architectures and training strategies. Four structures are used and all of them are based on deep transformer which are proven more effective than the standard Transformer-big models (Li et al., 2019).

In terms of data augmentation, we adopt R2L training (Zhang et al., 2019) to all the tasks. Monolingual data is only used in English → German task following the back-translation manner (Sennrich et al., 2016b). Different from the standard back-translation, we add noise to the synthetic source sentence in order to take advantage of large-scale monolingual text. In addition, we add a special token to the synthetic source sentence to help the model better distinguish the bilingual data and synthetic data. The in-domain finetuning (Sun et al., 2019) is very effective in our three experiments and specially, we propose a boosted finetuning method for English ↔ Chinese tasks. We also take advantage of the combination methods to further improve the translation quality. The "greedy search ensemble algorithm" (Li et al., 2019) is used to select the best combinations from single models. Then for English ↔ Chinese tasks we propose an iterative transductive ensemble (ITE) method based on the translation results of the ensemble models. For English → German task, we apply the noise channel model for re-ranking (Yee et al., 2019).

This paper was structured as follows: Section 2 describes the dataset. We present the detailed overview of our system in Section 3. The experiment settings and main results are shown in Section 4. Finally, we conclude our work in Section 5.

2 Dataset

2.1 Chinese ↔ English

The bilingual data used in Chinese ↔ English task includes all the available corpus provided by WMT2020: News Commentary v15, Wiki Titles v2, UN Parallel Corpus V1.0, CCMT Corpus, WikiMatrix, Back-translated news. The Chinese sentences are segmented by jieba segmentor[1] while the English side is processed by Moses tokenizer. We collect 18M sentence pairs after filtering.

2.2 English → German

The bilingual data used in this task includes all the available corpus provided by WMT2020. For the Paracrawl part, We filter most of the data due to

*Equal contribution. Correspondence to {*frostwu*, *brightxwang*, *vinnylywang*, *fangxuliu*}@*tencent.com*.

[1] https://github.com/fxsjy/jieba.

Proceedings of the 5th Conference on Machine Translation (WMT), pages 313–319
Online, November 19–20, 2020.

bad quality and collect 15M sentence pairs. Totally, 22M sentence pairs are used for training. Both the languages are tokenized by *tokenize.perl* script[2]. Then BPE is applied with 32K operations. The vocabulary is shared with 32K unique words. For monolingual data, we randomly select 80M sentences from NewsCrawl2017-2019 for back-translation and 45M are used for training after filtering

2.3 Data Processing

Pre-processing To pre-process the raw data, we apply a series of open-source/in-house scripts, including full-/half-width conversion, Unicode conversation, punctuation normalization, tokenization and true-casing. After filtering steps, we generated subwords via BPE (Sennrich et al., 2016c) with pre-defined merge operations of 32,000.

Filtering To improve the quality of data, we filtered noisy sentence pairs according to their characteristics in terms of language identification, duplication, length, invalid string and edit distance. More specifically, we filter out the sentences longer than 150 words. The word ratio between the source and the target must not exceed 1:1.3 or 1.3:1. According to our observations, the filtering method can significantly reduce noise issues including misalignment, translation error, illegal characters, over-translation and under-translation.

3 System Overview

3.1 Model Architecture

In our systems, we adopt four different model architectures with TRANSFORMER (Vaswani et al., 2017):

- **DEEP** TRANSFORMER (Dou et al., 2018; Wang et al., 2019; Dou et al., 2019) is the TRANSFORMER-BASE model with the 40-layer encoder.

- **HYBRID** TRANSFORMER (Hao et al., 2019) is the TRANSFORMER-BASE model with 40-layer hybrid encoder. The 40-layer hybrid encoder stacks 35-layer self-attention-based encoder on top of 5-layer bi-directional ON-LSTM (Shen et al., 2019) encoder.

- **BIGDEEP** TRANSFORMER is the TRANSFORMER-BIG model with 20 encoder layers.

- **LARGER** TRANSFORMER is similar to BIGDEEP model except that it uses 8192 as the FFN inner width.

The main differences between these models are presented in Table 1. To stabilize the training of deep model, we use the Pre-Norm strategy (Li et al., 2019). The layer normalization was applied to the input of every sub-layer which the computation sequence could be expressed as: normalize $\rightarrow$ Transform $\rightarrow$ dropout $\rightarrow$ residual-add. All models are implemented on top of the open-source toolkit Fairseq[3] (Ott et al., 2019).

3.2 Data Augmentation

Data augmentation is a commonly used technique to improve the translation quality. There are various of methods to conduct data augmentation such as back-translation (Sennrich et al., 2016a), joint training (Zhang et al., 2018) etc. In this section, we will introduce the methods we used in WMT2020.

3.2.1 Large-scale Back-translation

Back-translation is the most commonly used data augmentation technique to incorporate monolingual data into NMT (Sennrich et al., 2016a). The method first trains an intermediate target-to-source system, which is used to translate target monolingual corpus into source. Then the synthetic parallel corpus is used to train models together with the bilingual data.

In this work we apply the noise back-translations method as introduced in (Lample et al., 2018). When translating monolingual data we use an ensemble of two models to get better source translations. We follow (Edunov et al., 2018) to add noise to the synthetic source data. Furthermore, we use a tag at the head of each synthetic source sentence as Caswell et al. (2019) does. To filter the pseudo corpus, we translate the synthetic source into target and calculate a Round-Trip BLEU score, the synthetic pairs are dropped if the BLEU score is lower than 30. Notably, we only apply back translation to the English $\rightarrow$ German task. We find that back translation decrease the translation quality to Chinese $\leftrightarrow$ English tasks in our experiments.

[2]https://github.com/moses-smt/mosesdecoder/tree/master/scripts/tokenizer/tokenizer.perl

[3]https://github.com/pytorch/fairseq

	DEEP	HYBRID	BIGDEEP	LARGER
Encoder Layer	40	40	20	20
Decoder Layer	6	6	6	6
Attention Heads	8	8	16	16
Embedding Size	512	512	1024	1024
FFN Size	2048	2048	4096	8192

Table 1: Hyper-parameters of different Transformer models used in our system.

3.2.2 R2L Training

The approach is proposed by (Zhang et al., 2019). The main idea is to integrate the information of Right-to-Left (R2L) models to Left-to-Right (L2R) ones. Following this work, we translate the source sentences of the parallel data with both a R2L model and a L2R model, and use the translated pseudo corpus to improve the L2R model. We drop the pseudo parallel data if the BLEU score lower than 15. This method is applied to all the three tasks.

3.3 Finetuning

We use in-domain finetuning to further improve the model performance on news domain as previous study (Sun et al., 2019) shows that finetuning is very effective on the WMT2019 news translation tasks. For the three tasks, the finetuning is slight different and we will introduce them seprately in the following of this section.

Finetuning Zh → En Models For this task, we use all the previous development and test dataset as in-domain corpus D that includes WMT2017 development data, WMT2017 test data and WMT2018 test data. After training an NMT model M with the above methods, we finetune W on D with the same hyper parameters of training M. When testing on the WMT2019 test set, we achieve about 4-5 BLEUs improvement. As the in-domain corpus is very limited, we propose a boosted finetuning method by using the R2L training method to boost the finetuning process, which is named finetuning (boost). In our final submission, we add the WMT2019 test to D, the batch size is set to 2,048, the finetuning finished after 3k training steps.

Finetuning En → Zh Models We select the WMT2017 development data, WMT2017 test data and WMT2018 test data as the in domain corpus D in both tuning models and final submission which is different from Zh → En task. In addition, we do not use R2L training or add WMT2019 test to D, as we find this is useless. When finetuning, we reset the optimizer and use a fixed learning rate of 8e-5. The batch size is set to 1024 and the finetuning finishes after 900 upates.

Finetuning En → De Models We select the document whose source side is originally in English from all previous development and test dataset as in-domain corpus D. Single models are trained with the above methods are then finetune on D for one epoch with a fixed learning rate of 1e-4. In our final submission, the WMT2019 test set is added to D for better performance improvement.

3.4 Re-ranking

We use noisy channel model re-ranking method (Yee et al., 2019). This method is implemented in Fairseq [4]. Three features are used as following:

Source-to-Target Model Instead of a single model, we use the ensemble model as source-to-target model. Four well-trained single models are used. The decoding beam size is set to 25. We collect the log probability of each translation candidates.

Target-to-Source Model The target-to-source model is the channel mode which is used to translate the candidates back to source. We use a big transformer model for target-to-source.

Language Models For language model, we train a small GPT-2 model with FFN=8192 for target monolingual data.

Tuning We use random search to choose values in the range $[0, 3)$ for λ_1 , λ_2 and length penalty. The parameters are tuned on development set.

3.5 Ensemble

Model ensemble is a widely used technique in previous WMT workshops (Li et al., 2019; Sun et al.,

[4]https://github.com/pytorch/fairseq

2019; Wang et al., 2018a) which can boost the performance by combining the predictions of several models at each decoding step. In our work, we use two kinds of ensemble methods and finally the two are combined for further improvements.

3.6 Greedy Based Ensemble

This method is proposed by Li et al. (2019). The method adopts an easy operable greedy-base strategy to search for a better single model combinations on the development set. For more detail, please refer to the original paper. We also train single models with different hyper parameters to ensure the diversity. We refer to this method as **Ensemble** in the following.

3.7 Iterative Transductive Ensemble

Transductive ensemble (TE) is proposed by Wang et al. (2020c). The key idea is that source input sentences from the validation and test sets are firstly translated to the target language space with multiple different well-trained NMT models, which results in a pretranslated synthetic dataset. Then individual models are finetuned on the generated synthetic dataset. We propose an variation of TE, the Iterative Transductive Ensemble (ITE) which is based on Ensemble, as following:

Algorithm 1: Iterative Transductive Ensemble

Input: Single models M_1^m, In-domain corpus D, E_1^n is n different ensemble combinations
Output: Single models M_1^m

Translate D with E_1^n and get ${D'_1}^n$
Train each M_1^m on $D \cup {D'_1}^n$ and get ${M'_1}^m$, then $M_1^m = {M'_1}^m$
$t := 0$
while *not convergence* **do**
    Translate D with M_1^m and get ${D''_1}^n$, then ${D'_1}^n = {D'_1}^n \cup {D''_1}^n$
    Train each M_1^m on $D \cup {D'_1}^n$ and get ${M'_1}^m$, then $M_1^m = {M'_1}^m$
    $t := t + 1$
return ,

4 Experiments and Results

4.1 Setups

The implementation of our models is based on Fairseq [5]. All the single models are carried out on 8 NVIDIA V100 GPUs each of which have 32 GB memory. We use the Adam optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.98$. The gradient accumulation is used due to the high GPU memory consumption. The batch size is set to 8192 toknes per GPU and the "update-freq" parameter in Fairseq is set to 8. Specifically, for LARGE settings, the batch size is 4096 and "update-freq" is 16. We set max learning rate to 0.0007 and warmup-steps to 4000. All the dropout probabilities are set to 0.1. We select the checkpoint with the lowest loss on development set as the final checkpoint in each training. We calculate sacreBLEU score [6] for all experiments which is officially recommended. The WMT2019 testset (test2019) is used as the development set for all the tasks.

4.2 Chinese → English

Table 2 shows the Chinese → English translation results on validation set. We train multiple single models in each settings and report the best scores in Table 2. The R2L method can significantly improve the baseline by 2.45 BLEU scores. It is surprising to find a gain of almost 5 BLEU improvement on test2019 dataset. After we boost the in-domain corpus, we can achieve 1 more BLEU on the DEEP model. This illustrates that the finetuning is very effective on the WMT2019 test set.

In our experiments, the ensemble models consists of 5 single models: 1 HYBRID, 1 BIGDEEP, 3 LARGER models. As shown in the Table2, the ensemble models outperform the best single model by 1.06 BLEU score. We then apply transductive ensemble to LARGER models and finally the performance achieves 38.99. We also find that the single models that applied TE cannot bring further improvement to ensemble results. We do not apply re-ranking to this task, as we find that the improvement is insignificant. Our WMT 2020 Chinese → English submission achieves a SacreBLEU score of 36.8 and chrF score of 0.649.

4.3 English → Chinese

Table 3 shows the English → Chinese translation results on validation set. We also train multiple single models and report the best scores in the Table. After applying R2L method, we achieve 0.4 to 1 BLEU. We can observe that the improvement from finetuning is not as high as Chinese → English tasks, where only more 1 BLEU is gained. We also find that the boosted finetuning is harmful in this task, thus we omit the results. The ensemble

[5] https://github.com/pytorch/fairseq

[6] https://github.com/mjpost/sacrebleu

	DEEP	HYBRID	BIGDEEP	LARGER
Baseline	29.01	-	-	-
+R2L	31.46	31.42	32.07	32.41
+Finetuning	36.04	-	-	-
+Finetuning(boost)	37.02	37.23	37.38	37.62
Ensemble	38.68			
ITE	38.99			

Table 2: BLEU evaluation results on the WMT 2019 Chinese → English test set.

	DEEP	BIGDEEP	LARGER
Baseline	38.10	38.63	38.90
+R2L	39.09	39.01	39.31
+Finetuning	-	40.72	40.68
Ensemble	41.46		
ITE	42.26		

Table 3: BLEU evaluation results on the WMT 2019 English → Chinese test set.

	BIGDEEP	DEEP
Baseline	41.58	41.71
+R2L	43.05	42.73
+BT	44.37	44.06
+Finetuning	45.30	44.82
+Ensemble	45.7	
+reranking	45.9	
+PostProcessing	47.3	

Table 4: BLEU evaluation results on the WMT 2019 English → German test set.

setting consist of 4 models, that are 2 BIGDEEP, 2 LARGER models, which outperform the best single model by 0.74 BLEU.

4.4 English → German

Table 4 shows the results on English → German translation. The baseline is the BIGDEEP model using only bilingual data. R2L training boosts the BLEU score from 41.58 to 43.05. After adding back-translation, we further improve the BLEU score to 44.37. The finetuning can further achieve 0.93 BLEU improvement on the BIGDEEP model.

In this task, the ensemble models consists of 4 single models: 1 DEEP, 2 BIGDEEP, 1 LARGER models. As shown in Table 4, the ensemble models outperform the best single model by 0.4 BLEU score. We then apply noisy channel re-reranking to ensemble results and finally achieve 45.9 BLEU on the development set.

We apply a post-processing procedure. After translating the source-side, we normalize the English quotations appearing in the German translations to German-style quotations. We find this can improve the BLEU score on development set by 1.4 points.

5 Conclusion

This paper presents the Tencent Translation systems for WMT2020 Chinese → English news translation tasks. We investigate various deep architectures to build strong baseline systems. Then popular data augmentation methods such as back-translation and R2L training are used to improve the baselines. We also prove that in-domain finetuning is very effective for news translation tasks especially on Chinese → English task. Finally, we adopt the greed-based ensemble algorithm and propose an iterative transductive ensemble method for further improvement.

It is worth mentioning a number of advanced technologies reported in this paper are also adapted to our systems for biomedical translation (Wang et al., 2020b) and chat translation (Wang et al., 2020a) tasks, which respectively achieve up to 1st and 2nd ranks in terms of BLEU scores.

References

Isaac Caswell, Ciprian Chelba, and David Grangier. 2019. Tagged back-translation. *CoRR*, abs/1906.06442.

Zi-Yi Dou, Zhaopeng Tu, Xing Wang, Shuming Shi, and Tong Zhang. 2018. Exploiting deep representations for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4253–4262,

Brussels, Belgium. Association for Computational Linguistics.

Zi-Yi Dou, Xing Wang, Shuming Shi, and Zhaopeng Tu. 2019. Exploiting deep representations for natural language processing. *Neurocomputing*.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500, Brussels, Belgium. Association for Computational Linguistics.

Jie Hao, Xing Wang, Shuming Shi, Jinfeng Zhang, and Zhaopeng Tu. 2019. Towards better modeling hierarchical structure for self-attention with ordered neurons. In *EMNLP-IJCNLP*, pages 1336–1341.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Unsupervised machine translation using monolingual corpora only. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*.

Bei Li, Yinqiao Li, Chen Xu, Ye Lin, Jiqiang Liu, Hui Liu, Ziyang Wang, Yuhao Zhang, Nuo Xu, Zeyang Wang, Kai Feng, Hexuan Chen, Tengbo Liu, Yanyang Li, Qiang Wang, Tong Xiao, and Jingbo Zhu. 2019. The niutrans machine translation systems for wmt19. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 257–266, Florence, Italy. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. Fairseq: A fast, extensible toolkit for sequence modeling. *NAACL*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016c. Neural machine translation of rare words with subword units. In *ACL*.

Yikang Shen, Shawn Tan, Alessandro Sordoni, and Aaron Courville. 2019. Ordered neurons: Integrating tree structures into recurrent neural networks. In *ICLR*.

Meng Sun, Bojian Jiang, Hao Xiong, Zhongjun He, Hua Wu, and Haifeng Wang. 2019. Baidu neural machine translation systems for wmt19. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 374–381, Florence, Italy. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*.

Longyue Wang, Zhaopeng Tu, Wang Xing, Li Ding, Liang Ding, and Shuming Shi. 2020a. Tencent AI Lab machine translation systems for the WMT20 chat translation task. In *Proceedings of the Fifth Conference on Machine Translation*.

Mingxuan Wang, Li Gong, Wenhuan Zhu, Jun Xie, and Chao Bian. 2018a. Tencent neural machine translation systems for wmt18. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 526–531, Belgium, Brussels. Association for Computational Linguistics.

Qiang Wang, Bei Li, Jiqiang Liu, Bojian Jiang, Zheyang Zhang, Yinqiao Li, Ye Lin, Tong Xiao, and Jingbo Zhu. 2018b. The niutrans machine translation system for wmt18. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 532–538, Belgium, Brussels. Association for Computational Linguistics.

Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F Wong, and Lidia S Chao. 2019. Learning deep transformer models for machine translation. In *ACL*.

Xing Wang, Zhaopeng Tu, Longyue Wang Wang, and Shuming Shi. 2020b. Tencent AI Lab machine translation systems for the WMT20 biomedical translation task. In *Proceedings of the Fifth Conference on Machine Translation*.

Yiren Wang, Lijun Wu, Yingce Xia, Tao Qin, ChengXiang Zhai, and Tie-Yan Liu. 2020c. Transductive ensemble learning for neural machine translation. In *AAAI*, pages 6291–6298.

Kyra Yee, Yann Dauphin, and Michael Auli. 2019. Simple and effective noisy channel modeling for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5696–5701, Hong Kong, China. Association for Computational Linguistics.

Zhirui Zhang, Shujie Liu, Mu Li, Ming Zhou, and Enhong Chen. 2018. Joint training for neural machine translation models with monolingual data. *arXiv preprint arXiv:1803.00353*.

Zhirui Zhang, Shuangzhi Wu, Shujie Liu, Mu Li, Ming Zhou, and Tong Xu. 2019. Regularizing neural machine translation by target-bidirectional agreement. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 443–450.

Russian-English Bidirectional Machine Translation System

Ariel Xv [1]
Beihang University, Beijing 100191, China
zf1906251@buaa.edu.cn
Wenhan Chao [2]
Beihang University, Beijing 100191, China
chaowenhan@buaa.edu.cn

Abstract

This review depicts our submission to the WMT20 shared news translation task. WMT is the conference to assess the level of machine translation capabilities of organizations in the word. We participated in one language pair and two language directions, from Russian to English and from English to Russian. We used official training data, 102 million parallel corpora and 10 million monolingual corpora. Our baseline systems are Transformer models trained with the Sockeye sequence modeling toolkit, supplemented by bi-text data filtering schemes, back-translations, reordering and other related processing methods. The BLEU value of our translation result from Russian to English is 35.7, ranking 5th, while from English to Russian is 39.8, ranking 2th.

1 Introduction

We participated in WMT20 shared news translation task by building neural translation systems for one language pair and two language directions, from English to Russian and from Russian to English. Our systems are based on the framework of the Transformer neural machine translation model, using many techniques and approaches, including the use of BPE subword segmentation for open-vocabulary translation with a fixed vocabulary, large-scale back-translation,and model ensembling.

Neural machine translation(Bahdanau et al., 2014) has emerged as the most promising machine translation approach in recent years, showing superior performance on public benchmarks. The proposed attention mechanism brought a new revolution in the neural machine translation in most cases, making the overall effect of translation much better than before. Then, the Transformer(Vaswani et al., 2017) that makes full use of the attention mechanism demonstrated outstanding performance and effectiveness. Up to now, most of work uses the structure of Transformer, and its superiority has been widely recognized.

Since the beginning of machine translation research, the translation between Russian and English has been extensively developed.As early as 1954, Georgetown University in the United States under the IBM company completed the English-Russian machine translation experiment with the IBM-701 computer, which opened the prelude of machine translation research. During the period, there are three core technologies, rule-based machine translation, statistical machine translation and neural machine translation. However, as the application field of machine translation became more and more complex, the limitations of various technologies started to become obvious. Due to more application scenarios and higher requirements for accuracy, model optimization problems appeared.

The translation between Russian and English is extremely difficult because their linguistic features are distinguished and the lexical composition and grammatical structure of Russian are more complicated than those of English. In the early period,statistical machine translations were hoped to be implemented through phrase-based methods(Marcu and Wong, 2002) and related techniques for language models and translation models. These methods have solved the Russian-English translation problems to a certain extent. Yet, at the same time, there exists translation problems that are high time cost and poor translation effect.

Since then, the emergence of neural machine translation has brought new developments to Russian-English machine translation. The basic modeling framework for neural machine translation is an end-to-end sequence generation model, a framework and method for transforming input sequences into output sequences. There are two

Proceedings of the 5th Conference on Machine Translation (WMT), pages 320–325

points in the core part. One is to represent the input sequence through the encoder, and the other is to obtain the output sequence through the decoder. In addition, for machine translation, neural machine translation not only includes encoding and decoding, but also uses RNN(Sutskever et al., 2014) or other methods to encode sentence pairs. It also introduces an additional mechanism, the attention mechanism(Luong et al., 2015), to help us to convert sequences. These innovations lead to an increase in translation performance in comparison to earlier models. Later, Transformer appeared, which greatly enhances the neural machine translation performance.

This paper is based on Transformer, a neural machine translation network structure, to develop a two-way evaluation task between Russian and English. Taking into account the language characteristics of Russian and English, we have done appropriate operations in data preprocessing, including removing duplicates, deleting unreasonable sentence pairs, lowercase and Latinization operations, and judging sentence alignment problems, removing the parallel corpus with problems. The filtered parallel corpus is then sent to the model for training and the training results are tested.After getting the trained model, we start to consider using the back-translation operation to augment the data, continuing to filter the generated artificial corpus, and put it into the model training together with the original parallel corpus.

Finally, ensemble(Dietterich, 2000), average and rerank(Shen et al., 2004) operations are implemented on different models to improve the overall performance of the translation system.

2 Background

Neural network machine translation is based on a sequence-to-sequence overall structure consisting of an encoder and a decoder. The encoder converts the source language sentence into an intermediate sequence result, and the decoder converts the intermediate sequence result into a target language sentence. There is also the Attention mechanism to help make the results perform better. In the construction of the overall translation system, we used a lot of excellent methods proposed earlier in the literature.

The basic model used here is Transformer, introduced by(Vaswani et al., 2017) . The transformer is an attention-based structure proposed to deal with tasks that require sequence models, such as machine translation. Traditional neural machine translation mostly uses RNN or CNN as the model base of encoder-decoder, and Google's latest Attention-based Transformer model abandons the inherent formula and does not use any CNN or RNN structure. The model works in high-level parallel process, so training speed is also relatively fast while improving translation performance. But it is still computationally expensive.

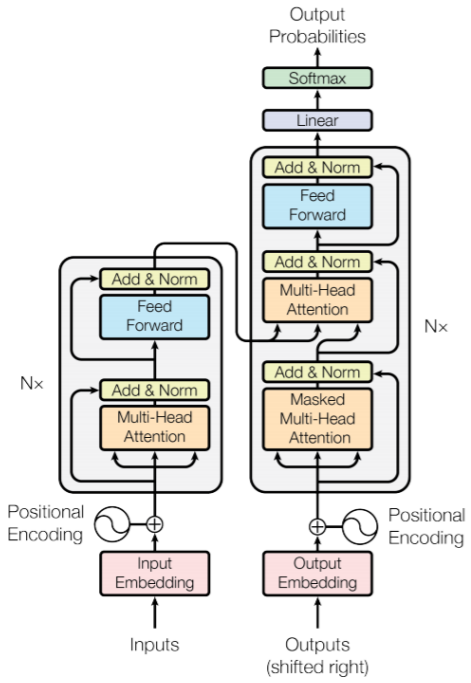


Figure 1: Transformer Structure.

The structure of Transformer is shown in Figure 1. The model is divided into two parts: the encoder and the decoder. The encoder is stacked by six identical layers, each with two more sub-layers. The first sub-layer is a long self-attention mechanism, and the second sub-layer is a simple fully connected feed forward network. A residual connection is added outside the two layers, and then layer normalization is performed. The output dimensions of all sub-layers and embedding layers of the model are d_{models}; the decoder also stacks six identical layers. However, in addition to the two layers in the encoder, the decoder also adds a third sub-layer, as shown in the figure which also uses the residual and layer normalization.

3 Data

We use all available bitext data which provided by WMT for the Russian-English language pair. For the monolingual data we use English and Russian Newscrawl as well as a filtered part of Commoncrawl in Russian. We choose to use Russian Commoncrawl to augment our monolingual data due to the relatively small size of Russian Newscrawl compared to English.

3.1 Data preprocessing

For the Russian-English language pair, we applied a series of preprocessing steps using scripts available in the Moses decoder(Koehn et al., 2007):

- replacing unicode punctuation,
- removing non-printing characters,
- normalizing punctuation,
- tokenization.

Also, we use joint byte pair encodings(BPE) with 32K split operations for subword segmentation(Sennrich et al., 2015) for each language.

3.2 Data Filtering

The large datasets which were crawled from the web would naturally be very noisy. And if they are used in their original and raw format, it may reduce the overall performance of the system. Clearning up these datasets is an important step to achieve good performance on any downstream tasks.

We applied two types of filters for data filtering: one is rule-based heuristics and another are filters based on language identification(Joulin et al., 2016).

For the Russian-English bitext data we used some data preprocessing methods to filter out them including:

- removing the bitext sentence pairs with a fixed length ratio above a certain threshold: for all the datasets we used a threshold of 3.
- removing sentence pairs with too short sentences: for all the sentences pairs we required a minimum number of five words.
- removing sentence pairs with too long sentences: we restricted all data to a maximum length of 100 words.

	En-Ru
No filter	112294588
+ length filter	102154821
+ langid filter	90826580

Table 1: Number of sentences pairs for different filtering schemes.

	En	**Ru**
Newscrawl	33600797	22348032
+ langid filter	32538613	20989583

Table 2: Number of sentences pairs for different filtering schemes.

Through observing the parallel data, we found that there is a surprisingly large amount of text segments in a wrong language in all provided parallel training data. So after some random inspection of the data, it is necessary to apply off-the-shelf language identifiers to the data for removing additional erroneous text from the training data. We apply language identification filtering called langid(Lui et al., 2012)which can classify each sentence in the parallel corpus.

So we can keep only sentence pairs with correct languages on both sides. At last, we filter out about 15% of the original parallel data. See Table 1 for details on the bitext dataset sizes.

For the monolingual English and Russian Newscrawl data we also apply langid filtering. As the monolingual Newscrawl data for Russian is relatively smaller than that of English, we have to augment the Newscrawl data for Russian with monolingual data from commoncrawl corpus.But there is a problem that the quality of commoncrawl corpus is very poor but is also noisy.

4 Experiment

For this evaluation task, we first start from the data preprocessing, through data expansion operations to obtain the data that needs to be trained, and then input the Transformer model for training. We test the training results and finally ensemble results according to the model generated by different strategies, average and rerank operations,for the best results. Next, the specific experiment content will be presented separately. The overall project process is showed in Figure 2.

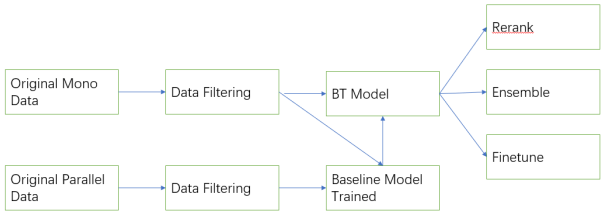


Figure 2: Project Process.

4.1 Base System

Our base system is based on the Transformer architecture (Vaswani et al., 2017) as implemented in Sockeye(Hieber et al., 2017). Due to the time cost and hardware cost of the evaluation task, we choose the basic version of Transformer. The encoder and decoder respectively have 6 sub-layers and the multi-head attention mechanism has 8 heads. The word embedding vector size is 512. We trained all our models using MXNET, which is the deep learning library that Amazon chose. The parameters setting of our models are listed in Table 3.

After the above processing, we use the parallel corpus which provided by the task organizers and direct it into the model for training and testing. The results of the base model can be used to generate reverse translation data to augment the corpus and continue training. The purpose is to maintain the generalization ability and robustness of the model to the greatest extent, and to provide reference for other model training results.

4.2 Large-scale Back-translation

Back-translation(Edunov et al., 2018) is an effective method to improve neural machine translation with monolingual data. It can incorporate monolingual data into a translation system. Firstly,we trained a baseline model that is used to translate monolingual target data into additional synthetic parallel data. This data is used in conjunction with original bitext data the desired source-to-target system.

In this work, due to the training time cost limitation we respectively only selected 10 million Russian and English sentences from the official monolingual corpus for back translation operations. We used back-translations obtained by beam-search(Edunov et al., 2018) from an ensemble of two target-to-source models. We adopt the method to tune the amount of bitext and pseudo-parallel corpora the model is trained on. We found that a ratio of 1:1 synthetic to bitext data can perform the best.

4.3 Fine-tuning

Fine-tuning is a common and effective method to improve machine translation quality especially for a downstream task. When we complete training on the original bitext and pseudo-parallel data, we train an special epoch on a smaller domain-specific data. It can make the model more sensitive to specific domain scenation and then get better results. Here, we select a corpus with much similarity to the test set from the training set to fine-tune the trained model. The similarity scores between the test corpus and the training corpus are sorted and ranked. Then the parallel sentence pairs with higher scores are found and the corpus is extracted as a fine-tuning corpus. In this way, about 5,000 pieces of data are obtained and this part of the corpus is input into the previously trained model to obtain the result of fine-tuning the model, so that it can perform better on the test set.

4.4 Model Reranking

N-best reranking is a method of improving translation quality by scoring and selecting a candidate hypothesis from a list of n-best hypotheses generated by a trained model. Extracting only one of the highest-scoring statements from the translation results of the model as an output is not necessarily the best result. So this strategy can be used to extract the best three from each translation model result as a candidate set. Then use some rules to rerank and get the best one as the output result. The translated content thus obtained is the comprehensive output of multiple results of each model. The rules used here include weighted summation of beam search score and the language model scores. The first one is based on the beam score returned during decoding, but different models have different performances, so it is difficult to sort under a uniform metric. So we introduced different weights for different models. Using beam score weight as the final score for each translation result, the final result was obtained by screening. The second one gives scores of the generated translations using the pre-trained language model. They are judged from the linguistics itself and the sentences with the highest scores are selected. The final result is an output that combines the highest scores of the two methods described above.

The above models also had different batch sizes,

Parameters	Transformer
optimizer	adam
max-num-checkpoint-not-improved	16
num_words	50000:50000
optimized-metric	perplexity
max-seq-len	100:100
loss	cross-entropy

Table 3: The parameters setting of Transformer are implemented by Sockeye .

comparison of the number of graphics cards and vocabulary sizes in the training process. We extracted them for the optimal results. Finally, the output is simply post-processed. In order to comply with common practice in natural language processing. However, due to the limitations of time and hardware resources, not every experiment has been refined and detailed totally, so there is still improvement of results in the future.

4.5 Ensemble Model

Ensemble is a method that combines the results of multiple models. The purpose of this is to complement the advantages of different models, make up for the problems that fall into the local optimum and get the results of the machine translation model with better comprehensive effects. For the sake of simplicity, only different initialization random seed parameters are set for the same model. So training of multiple models is performed, generally two or three models, and finally the results of all models are subjected to ensemble operation. By composing and complementing multiple models, we obtain the comprehensive optimal results of data translation.

5 Results

Results and ablations from Russian to English are shown in Table 4, from English and Russian are shown in Table 5. We report case-sensitive SacreBLEU scores using SacreBLEU(Post, 2018). We report all the case-sensitive BLEU(Koehn et al., 2007) score of our submitted system on this year's test set.

5.1 Russian To English

From Russian to English, we can see that langid filtering and ensembling improve our baseline performance on this year's test set by about 0.7 BLEU. This is perhaps due to the addition of higher quality bitext data and improved data filtering techniques. The addition of back-translated(BT) data

Type of Text	Pair	Bleu	Improve
base-re	RU-EN	33.1	0
filter-re	RU-EN	34.2	+1.1
ensemble-re	RU-EN	36.6	+2.4
fintune-re	**RU-EN**	**39.1**	+2.5
rerank-re	RU-EN	38.2	-1.1

Table 4: Russian-English Experiment Result.

Type of Text	Pair	Bleu	Improve
base-re	EN-RU	23.1	0
filter-re	EN-RU	24.2	+1.1
ensemble-re	EN-RU	24.5	+0.3
fintune-re	**EN-RU**	**24.8**	**+0.3**
rerank-re	EN-RU	24.6	-0.2

Table 5: English-Russian Experiment Result.

improves single model performance by about 0.3 BLEU, combining this with fine-tuning and ensembling gives us a total of 3 BLEU. We composed two models which have different random seeds and then re-trained on the fine-tuning corpus. Finally, applying reranking on top of these strong ensembled systems gives another 1.4 BLEU.

5.2 English To Russian

From English to Russian, we observe similar trends to Russian to English, with langid filtering and ensembling improving performance of a baseline system by 1.6 BLEU. Back-translatoin adds 1.5 BLEU, again mostly likely due to the lower quality bitext data available. Also we composed two models which have different random seeds and then re-trained on the fine-tuning corpus. Fine-tuning, ensembling, and reranking add almost 3 BLEU, with reranking contributing 1.2 BLEU.

6 Conclusions

This paper describes our submission to the WMT20 news translation task. In the evaluation task, we es-

tablished a Russian-English bidirectional machine translation system based on Transformer. For translations between Russian and English, we use the same strategy of filtering bitext data, performing beam-search back-translation on monolingual data. Then we train strong individual models on a combination of this data. Each of these models is fine-tuned and ensembled into a final system that is used for decoding with model reranking. In the final list, we got 2th in Ru-En, and 5th in En-Ru. Good results have been obtained in limited time and hardware resources, which is also in line with the industry's demands for service construction. In the whole experiment process, we also gained a lot of experience in data processing and experimental design, which will be of great help in later research and study. We will continue to improve the previous experiments, strive to get better results, and see what rankings can eventually be achieved, in preparation for the next year.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Thomas G Dietterich. 2000. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pages 1–15. Springer.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. *arXiv preprint arXiv:1808.09381*.

Felix Hieber, Tobias Domhan, Michael Denkowski, David Vilar, Artem Sokolov, Ann Clifton, and Matt Post. 2017. Sockeye: A toolkit for neural machine translation. *arXiv preprint arXiv:1712.05690*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*, pages 177–180. Association for Computational Linguistics.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

Daniel Marcu and Daniel Wong. 2002. A phrase-based, joint probability model for statistical machine translation. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, pages 133–139.

Matt Post. 2018. A call for clarity in reporting bleu scores. *arXiv preprint arXiv:1804.08771*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Libin Shen, Anoop Sarkar, and Franz Josef Och. 2004. Discriminative reranking for machine translation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 177–184.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

The DeepMind Chinese–English Document Translation System at WMT2020

Lei Yu*, Laurent Sartran*, Po-Sen Huang*, Wojciech Stokowiec*, Domenic Donato*
Srivatsan Srinivasan*, Alek Andreev*, Wang Ling*, Soňa Mokrá, Agustin Dal Lago
Yotam Doron, Susannah Young, Phil Blunsom, Chris Dyer*
DeepMind
{leiyu, lsartran, posenhuang, wstokowiec, domenicd,
vatsan, alekandreev, lingwang, sonka, agudallago,
ydoron, susannahy, pblunsom, cdyer}@google.com

Abstract

This paper describes the DeepMind submission to the Chinese→English constrained data track of the WMT2020 Shared Task on News Translation. The submission employs a noisy channel factorization as the backbone of a document translation system. This approach allows the flexible combination of a number of independent component models which are further augmented with back-translation, distillation, fine-tuning with in-domain data, Monte-Carlo Tree Search decoding, and improved uncertainty estimation. In order to address persistent issues with the premature truncation of long sequences we included specialized length models and sentence segmentation techniques. Our final system provides a 9.9 BLEU points improvement over a baseline Transformer on our test set (newstest 2019).

1 Introduction

The WMT2020 Shared Task on translating news data from Chinese into English provides a challenging test for machine translation systems and an ideal domain for researchers to evaluate new techniques. The DeepMind submission to the constrained data track is based on the modular noisy channel document translation architecture advocated by Yu et al. (2020). In this formulation, the posterior probability of a translation is the product of the unconditional probability of the output document (the language model) and the conditional probability of the translation from the output to source (the channel model). By assuming sentences within a document are independently translated, we can train the channel model using readily available parallel sentences, rather than being reliant on less numerous parallel documents, and the language model on monolingual documents. This modular approach allows the components of the system to be implemented and optimized independently while at inference time, when we reason over the posterior distribution of translations given the source document, conditional dependencies between translations are induced by the language model prior.

The core of our document-level translation architecture is the noisy channel reranker. It requires proposal, channel, and language models, each of which is optimized separately using different techniques and approaches. For the proposal and channel models we use Transformer models (Vaswani et al., 2017) (§4.1) with data augmentation (§4.2), such as back translation (Edunov et al., 2018), distillation (Kim and Rush, 2016; Liu et al., 2016), and forward-translated parallel documents. We further improve these sequence-to-sequence (seq2seq) models by fine-tuning them with in-domain data (§4.3). To improve the robustness of the reranker we apply adversarial training and contrastive learning methods for uncertainty estimation (§4.4). Finally, we include candidate translations generated by Monte-Carlo Tree Search (MCTS) (§B) in order to improve the diversity of the candidate pool for the reranker. Our language models are based on the Transformer-XL architecture (Dai et al., 2019) and optimized with distillation and fine-tuning with in-domain data (§5).

During development, we observed weaknesses in our system's translations for long sentences, largely due to premature truncations. We developed several techniques to mitigate this issue such as sentence segmentation (breaking sentences into logical complete segments) and training specialized models with synthetically constructed long sequences to generate additional proposals for our reranker (§A).

Experiments show that the aforementioned techniques are very effective: our system outperforms the Transformer baseline by 9.9 BLEU points on our test set (newstest2019). Our final system achieves a BLEU score of 35.4 on the

*Equal contribution.

Proceedings of the 5th Conference on Machine Translation (WMT), pages 326–337
Online, November 19–20, 2020.

Chinese→English news test set of WMT2020.

2 Document Translation via Bayes' Rule

Following Yu et al. (2020), we model document translation via Bayes' rule. We define $\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_I)$ as the source document with I sentences, and similarly, $\boldsymbol{Y} = (\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_J)$ as the target document with J sentences, where $\boldsymbol{x}_i$ and $\boldsymbol{y}_j$ denote the ith sentence in the source document and the jth sentence in the target document respectively. We assume that $I = J$.

The translation of a document $\boldsymbol{X}$ is determined by finding the document $\hat{\boldsymbol{Y}}$, where $p(\hat{\boldsymbol{Y}} \mid \boldsymbol{X})$ is maximal.

$$\begin{aligned}\hat{\boldsymbol{Y}} &= \arg\max_{\boldsymbol{Y}} p(\boldsymbol{Y} \mid \boldsymbol{X}) \\ &= \arg\max_{\boldsymbol{Y}} \underbrace{p(\boldsymbol{X} \mid \boldsymbol{Y})}_{\text{channel model}} \times \underbrace{p(\boldsymbol{Y})}_{\text{language model}} . \end{aligned} \quad (1)$$

We further assume that sentences are independently translated, and that the sentences within a document admit a left-to-right factorization according to the chain rule. Therefore, we have

$$\hat{\boldsymbol{Y}} \approx \arg\max_{\boldsymbol{Y}} \prod_{i=1}^{|\boldsymbol{Y}|} p(\boldsymbol{x}_i \mid \boldsymbol{y}_i) \times p(\boldsymbol{y}_i \mid \boldsymbol{Y}_{<i}), \quad (2)$$

where $\boldsymbol{Y}_{<i} = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_{i-1})$ denotes a document prefix consisting of the first $i-1$ target sentences.

The advantages of this formulation is that during training the translation models can be learned from parallel sentences and monolingual documents which are vastly available in practice unlike parallel documents. During test time, when a source document is observed, conditional dependencies between the translation of the source sentences are created in the posterior.

2.1 Reranking

Because of the global dependencies in the posterior distribution, decoding in the aforementioned document translation model is computationally expensive. Following Yu et al. (2020), we use an auxiliary proposal model $q(\boldsymbol{y} \mid \boldsymbol{x})$, that approximates the posterior distribution using a direct model, to focus our search on promising parts of the output space. We then carry out the reranking process using an iterative beam search, over candidates generated by the proposal model q, to optimize the objective:

$$\begin{aligned}\mathcal{O}(\boldsymbol{X}, \boldsymbol{Y}_{<i}, \boldsymbol{y}_i) = {} & \lambda_1 \log p_{\text{PM}}(\boldsymbol{y}_i \mid \boldsymbol{x}_i) + \\ & \lambda_2 \log p_{\text{AM}}(\boldsymbol{y}_i \mid \boldsymbol{x}_i) + \\ & \lambda_3 \log p_{\text{CM}}(\boldsymbol{x}_i \mid \boldsymbol{y}_i) + \\ & \log p_{\text{LM}}(\boldsymbol{y}_i \mid \boldsymbol{Y}_{<i}) + \\ & \lambda_4 |\boldsymbol{y}_i| + \\ & \mathcal{O}(\boldsymbol{X}, \boldsymbol{Y}_{<i-1}, \boldsymbol{y}_{i-1}), \end{aligned} \quad (3)$$

where p_{PM} is the proposal probabilities model, p_{AM} is the adversarially trained proposal model (§4.4.1), p_{CM} is the channel model (§4.4.2), p_{LM} is the language model (§5.1), and $|\boldsymbol{y}|$ denotes the number of tokens in the sentence $\boldsymbol{y}$. The weights of component models (λs) are hyperparameters to be tuned in experiments.

In practice, we generate for each source sentence $\boldsymbol{x}_i$ in the document $\boldsymbol{X}$, a series of *candidates* $\boldsymbol{y}_i$, using the proposal model q. As all of the terms in the objective, except for p_{LM}, only involve independent target sentences, they can be computed ahead of time in a *scoring* phase. The *scored candidates* are then passed to the reranker, where the language model is evaluated on the successive prefixes explored by the search, and which outputs the final document $\hat{\boldsymbol{Y}}$.

Iterative beam search The algorithm starts with k complete documents using randomly selected candidates for each of the source sentences. We then iterate through every source sentence $\boldsymbol{x}_i$, replacing the randomly picked initial candidate with every available candidate $\boldsymbol{y}_i$. We pick the top k scoring *complete* documents and continue iterating over the document. Unlike traditional beam search used by Yu et al. (2020), we go through every sentence in the document multiple times, until the top 1 translation converges (usually 2 to 4 full iterations). This allows for context from latter sentences in the document to inform the choice of earlier candidates.

Iterative beam search found improvements in the model objective over traditional beam search in 63% of the documents in our test set. Improvements in objective did not translate in a stable improvement in BLEU or META scores (Eqn. 4) – in fact those scores were slightly reduced for a number of documents. Nevertheless, an informal human evaluation of translated documents showed preference for iterative beam search.

Selection of the hyperparameters λ We perform a grid search over the hyperparameters λ to

maximize a metric on the validation set. The metric we use is the following META score, combining corpus-level BLEU, TER, METEOR, and the 0.1-quantile of per-document BLEU, such that:

$$\begin{aligned}(1-\text{META})^4 = \text{TER}\times \\ (1-\text{BLEU})\times \\ (1-\text{METEOR})\times \\ (1-q_{0.1}(\text{BLEU})).\end{aligned} \quad (4)$$

When several configurations of hyperparameters achieve values of META very close to the maximum (within 0.02), we pick the one maximizing BLEU and/or minimizing the L_2 norm of the λs, considered as a vector. This corresponds to an intuitive prior towards giving more weight to the language model.

3 Training Data

To train all of the models used in our system, we made use only of the constrained data provided to shared task participants. In this section, we discuss the preprocessing and normalization techniques we carried out in an attempt to reduce spurious uncertainty in the modeling problem.

Text preprocessing We carried out the following text normalization steps prior to use in any models:

- Text normalization. Unicode canonicalization (NKFD from), replacement of common multiple encoding errors present in training data, standardization of quotation marks into "directional" variants, conversion of any traditional Chinese characters into simplified forms. Replacement of non-American spelling variants with American spellings using the aspell library.[1]
- Segmentation into words. Chinese was segmented into word-like units using the Jieba segmentation tool.[2] Punctuation was split from English words using a purpose-built library. These processes were not completely invertible, but they could be undone with simple rules so as to generate presentation-ready English and Chinese.
- True-casing. Words containing only an initial capital letter that occurred at the start of a sentence were replaced with the capitalized variant that occurred most frequently in other positions of the English monolingual training data. Thus, in the previous sentence the initial token would have been *words* rather than *Words*.

Subword units To encode text into sub-word units, we used the `sentencepiece` tool (Kudo and Richardson, 2018). For seq2seq models (i.e., the channel model and proposal models), we trained the segmentation model on the first 10 million sentences of the parallel training corpus,[3] using joint source and target unigram (Kudo, 2018) subword segmentation algorithm with a target vocabulary of 32K tokens and minimum character coverage of 0.9995, which resulted in 32,768 word pieces.[4] For the language model, we used the English side alone with the same vocabulary size and a character coverage of 1.0.

4 Proposal and Channel Models

The proposal model, used to generate candidate translations, and the scoring models (proposal probability model, adversarially-trained proposal model, channel model), used to compute features for the reranker, are seq2seq models. We describe here how we train and use them.

4.1 Sequence-to-Sequence Model

All our models are based on the Transformer architecture (Vaswani et al., 2017). We increased the inner dimension of the feed-forward network from 4,096 to 8,096 and decreased the model size (d_{model}) from 1,024 to 512, which allowed us to use 12 layers with 16 attention heads each. Additionally, we tied the source and target embedding layers. Following (Vaswani et al., 2018), we applied layer normalization to the input of every sublayer as opposed to its original placement after the element-wise residual addition. We used different dropout values for different components: 0.1 for the multi-head attention, 0.05 in the feed-forward network, and finally 0.3 after the sub-layer. Learning rate schedule and dropout were found using the Batched Gaussian Process Bandits (Desautels et al., 2014) algorithm as implemented by Vizier (Golovin et al., 2017). All other hyperparameters

[1] `http://wordlist.aspell.net/varcon-readme/`

[2] `https://github.com/fxsjy/jieba`

[3] NC followed by CWMT, WikiTitles and UN.

[4] We tried both larger vocabulary sizes and separate vocabularies but neither of these led to an improvement for our system.

were decided upon using grid search. During training, we used a maximum sequence length of 96. For decoding, we used beam search with beam size 6, and set the length penalty alpha to 0.8, and a maximum decoding length of 384. Multi model ensembling was done via softmax output averaging as described in (Freitag et al., 2017).

4.2 Data Augmentation

In this section, we introduce how we augment data based on the given bilingual data and monolingual data. When we train the proposal and channel models, we use all the augmented data along with the original bilingual data.

Back-translation We perform back-translation from monolingual English data using fine-tuned channel models (English→Chinese) with top-k sampling following (Edunov et al., 2018) with $k = 50$ during decoding. We used the same in-domain monolingual data as described in §5.2. We score the back-translated data with fine-tuned proposal (Chinese→English) models, and filter them based on the quantiles of length ratios, sequence log-probability and cross-entropy between one-hot empirical translations and logits from the scorer model. The filtering helped to reduce the size of data from 43.4M to 29.9M paired sentences.

Forward translation to generate synthetic parallel documents We applied a version of our system to monolingual Chinese documents from Gigaword to get synthetic English documents. We only kept documents having between 4 and 25 sentences, we rejected outliers according to their probabilities under the language model, the channel model, and to the overall objective. These were then used to train subsequent versions of the forward (Chinese→English) models.

Data distillation We use knowledge distillation (Kim and Rush, 2016) to do distillation on the original dataset. Specifically, we translate the source-side of the bilingual data using previously trained proposal models (including Right-to-Left (Liu et al., 2016) and Left-to-Right models) and generate distilled candidates. The generated sentences are filtered if BLEU scores are below 30 (Wang et al., 2018; Sun et al., 2019). We then train models on the filtered data along with the original bilingual data and back-translation data. We repeat this process three times using models trained on newly generated data from the previous iteration. We empirically do not find Right-to-Left models significantly differ from Left-to-Right models in performance. Qualitatively we find that distilled data correct few errors in the original bilingual data.

4.3 Fine-tuning

Fine-tuning with in-domain data has been an effective approach for improving translation quality as shown by existing work (Sun et al., 2019; Ng et al., 2019). After training the proposal models with the mix of real and synthetic parallel data, we fine-tuned the models with CWMT and a subset of *newstest2017* and *newstest2018* which were not used for validation.

4.4 Improving Uncertainty Estimation

To improve the robustness of noisy channel reranking, we explore two approaches for improving uncertainty estimation of the seq2seq scoring models.

4.4.1 Adversarially Trained Proposal Models

To simulate different wordings and noises in source and candidate sentences, we follow Cheng et al. (2019) to train the models on noisy adversarial inputs and targets. We use bidirectional language-models to provide the noisy candidates and select the candidates with highest loss (i.e., adversarial source-target inputs). During the training, we optimize the original loss with clean source-target pairs, the language model losses for source and target sides, and the adversarial loss using adversarial source-target inputs. In the final scoring, we use an ensemble of eight adversarially trained models with few differences from Cheng et al. (2019): (a) We explore training with and without the language model losses. Though the models trained without the language model loss generate quite noisy sentences, we empirically find this approach still helps the overall performance. (b) In addition to using the clean hard-labels for the noisy source-target pairs for the adversarial loss as in the original work, we explore a variation using a KL loss between the adversarial source-target logits and the clean source-target logits. We find this variant also improves the overall performance.

4.4.2 Contrastive Channel Models

When scoring candidates, we want the channel models to be sensitive to translation noise (dropped words, permutation, or blanked words) (Edunov et al., 2018). Hence, we develop contrastive training (Yang et al., 2019; Welbl et al., 2020) to train

the models such that it will be more robust in estimating the channel probabilities. Specifically, we use n-gram Transformers (Chelba et al., 2020) with the contrastive loss:

$$\max\left\{\log p(\tilde{\boldsymbol{x}} \mid \boldsymbol{y}) + \eta - \log p(\boldsymbol{x} \mid \boldsymbol{y}), 0\right\}, \quad (5)$$

where $\tilde{\boldsymbol{x}}$ denotes a noisy version (random word deletion, blank, or permutation) of $\boldsymbol{x}$, and $p(\tilde{\boldsymbol{x}} \mid \boldsymbol{y})$ is the perturbed loss term. We ensemble 8 models with a few variants for final channel model scoring. These variants consist of the followings: (1) We use $\eta = \{0.01, 0.001\}$. (2) We use n-gram transformer with $n = 2, 8$. (3) We use models with perturbing source sentences $p(\tilde{\boldsymbol{x}} \mid \boldsymbol{y})$ and models with perturbing target sentences $p(\boldsymbol{x} \mid \tilde{\boldsymbol{y}})$. (4) Instead of using the contrastive loss, we include two models trained to minimize the perturbed loss terms directly. (5) Unlike Yang et al. (2019), where the authors firstly train with the maximum likelihood objective and then finetune with the contrastive loss, we find it empirically works better to train models with linearly increased weights (increasing from 0 to 1 during training) to the contrastive loss (Eq. (5)) along with the original negative log likelihood loss.

4.5 Filtering Candidate Translations

After obtaining candidate translations from strong proposal models, we filter out candidates with length ratio outside of $[e^{-1}, e^{1}]$, or which do not end with end-of-sentence punctuation when the source does, or with more than 4 consecutive identical tokens, or which are excessively compressible, indicating repeated contents, according to the following. We learn a piece-wise linear ordinary least squares model of the zlib-compressed length of true English sentences from their uncompressed length in UTF-8, using the English side of the training data. We then reject candidates the actual compressed length of which is more than 12 standard deviations below their predicted compressed length.

5 Language Model

In this section, we describe the architecture of the language models we used and how we trained them.

5.1 Model

The auto-regressive document language model is a Transformer-XL (Dai et al., 2019), with attention memory length of 512. Following Rae and Razavi (2020), we also used 4-layer blocks of short and long (128-128-128-512) attention memories, capturing short-range correlations in the earlier layers and long-range correlations in the later ones. This led to a 20% speedup of training, and helped the model generalize better to the validation set. We also used knowledge distillation in our Transformer-XL model with a setup similar to Born Again Neural Networks (BANN) (Furlanello et al., 2018), where we regularize the original loss function with term based on the cross-entropy between the new models outputs (student) and the outputs of the original (teacher) model.

Model	Train Data	Fine-tuning	PPL
Transformer-XL	Raw	No	29.4
Transformer-XL	In domain	No	27.4
+ memory + BANN	In domain	No	26.7
+ memory + BANN	In domain	Yes	24.3

Table 1: Language model perplexities per token on the validation set

Let $\mathcal{L}$ denote cross entropy loss function, $\boldsymbol{y}$ one-hot encoded label, $\boldsymbol{s}$ and $\boldsymbol{t}$ outputs of the student and teacher model respectively, then the BANN loss is defined as follows:

$$\mathcal{L}_{\text{BANN}} = \sum_{i=1}^{T} \mathcal{L}(\boldsymbol{y}_i, \boldsymbol{s}_i) \; + \lambda \cdot \mathcal{L}(\boldsymbol{t}_i, \boldsymbol{s}_i). \quad (6)$$

We trained our student network on the loss function in Eqn. 6 and found that $\lambda = 1$ had the best validation perplexity.

5.2 Data

The English data used to train our language models was prepared as described in §3.

In-domain document data for LM training We found that training LMs on a subset of training data that was more closely aligned with the validation set vastly improved the perplexity on the validation and test sets ($\approx 10\%$). To select a well-aligned subset of training data, we ranked the training data according to TF-IDF similarity with each validation document and collected the top 1,000 documents for each validation query together, to form our training data for the LM training. We also tried mixing this sub-sampled in-domain data with the raw data using different weights (essentially equivalent to up-weighting the in-domain data) and found that using purely in-domain data outperformed all other mixing schemes in terms

System	BLEU
Big Transformer	28.1
+ Data augmentation (§4.2)	33.6
+ Fine-tuning (§4.3)	35.8
+ Ensembling (§4.1)	36.6
+ Reranking (§2.1)	37.2
+ Length-targeting improvements (§A)	38.0

Table 2: SacreBLEU scores on newstest2019 Chinese-English.

of held-out perplexities and thus, the in-domain data became our training dataset. As an auxiliary benefit, the model trained on the in-domain dataset (340K iterations) also converged much earlier than the one trained on the raw dataset (500K iterations).

Similar to the sequence model fine-tuning outlined in §4.3, we also fine-tuned our trained language model in order to align the model more closely with the language constructs and domain information in our test data. Table 1 shows the perplexity numbers on the validation set obtained by different train data and model variants described above on the validation dataset.

6 Experiments and Results

We use the original Chinese subset of *newstest2017* and *newstest2018* as our validation set and *newstest2019* as our test set.

The candidate translations for the reranker are generated by 8 ensemble models (6 from each).

Table 2 presents the results of our models on the test set. We report case-sensitive SacreBLEU scores (Post, 2018). Both data augmentation and fine-tuning significantly improve the performance. Ensembling and noisy channel reranking gives about 0.8 and 0.6 BLEU boost, respectively. Finally, our specialized methods for handling long sequences (described in §A) yield a further 0.8 BLEU improvement.

In our final submitted system, we tune the weights of component models and the hyperparameters of sentence segmentation models using a combination of our validation set and test set. For the candidate translations of the reranker, apart from the existing 48 proposals generated by 8 ensemble models, we include additional 48 proposals generated by 8 ensemble models which are fine-tuned with CWMT, newstest2017, newstest2018, and newstest2019. We also include translations generated by MCTS decoding (§B) in our non-primary system. We find that adding a feature marking the length of source sentences longer than 60 words helps the reranker handle long sentences better. We therefore include this feature in addition to proposal probability, adversarial proposal probability, channel probability, language model probability, and length bonus (Eqn. 3).

Our system achieves a 35.4 BLEU score on newstest2020.

7 Conclusion

This paper describes the DeepMind submission to the WMT2020 news Chinese-English translation task. Using the noisy channel model (Yu et al., 2020) as our core document translation system, we optimized its component models using data augmentation, fine-tuning with in-domain data, MCTS decoding (§B), and knowledge distillation. We also addressed premature termination in long sentences by training specialized length expert models and segmenting long sentences into multiple shorter sentences (§A). We have demonstrated the marginal contributions of these methods in our analysis and our final system comprising all these methods outperforms the Transformer baseline by 9.9 BLEU points on newstest2019.

References

Ciprian Chelba, Mia Chen, Ankur Bapna, and Noam Shazeer. 2020. Faster transformer decoding: N-gram masked self-attention. *arXiv preprint arXiv:2001.04589*.

Yong Cheng, Lu Jiang, and Wolfgang Macherey. 2019. Robust neural machine translation with doubly adversarial inputs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4324–4333.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G. Carbonell, Quoc Viet Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2978–2988. Association for Computational Linguistics.

Thomas Desautels, Andreas Krause, and Joel Burdick. 2014. Parallelizing exploration-exploitation tradeoffs in gaussian process bandit optimization. *Journal of Machine Learning Research (JMLR)*, 15:40534103.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 489–500.

Markus Freitag, Yaser Al-Onaizan, and Baskaran Sankaran. 2017. Ensemble distillation for neural machine translation. *arXiv preprint arXiv:1702.01802*.

Tommaso Furlanello, Zachary Chase Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. 2018. Born-again neural networks. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 1602–1611. PMLR.

Daniel Golovin, Benjamin Solnik, Subhodeep Moitra, Greg Kochanski, John Karro, and D. Sculley. 2017. Google vizier: A service for black-box optimization. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1487–1495. ACM.

Mark Hopkins and Jonathan May. 2011. Tuning as ranking. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1352–1362, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327, Austin, Texas. Association for Computational Linguistics.

Levente Kocsis and Csaba Szepesvri. 2006. Bandit based monte-carlo planning. In *ECML*, volume 4212 of *Lecture Notes in Computer Science*, pages 282–293. Springer.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Brenden M. Lake and Marco Baroni. 2018. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 2879–2888. PMLR.

Lemao Liu, Masao Utiyama, Andrew Finch, and Eiichiro Sumita. 2016. Agreement on target-bidirectional neural machine translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 411–416.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook fair's wmt19 news translation task submission. *arXiv preprint arXiv:1907.06616*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers, WMT 2018, Belgium, Brussels, October 31 - November 1, 2018*, pages 186–191.

Jack W. Rae and Ali Razavi. 2020. Do transformers need deep long-range memory? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7524–7529. Association for Computational Linguistics.

Meng Sun, Bojian Jiang, Hao Xiong, Zhongjun He, Hua Wu, and Haifeng Wang. 2019. Baidu neural machine translation systems for WMT19. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 374–381.

Ashish Vaswani, Samy Bengio, Eugene Brevdo, Francois Chollet, Aidan Gomez, Stephan Gouws, Llion Jones, Łukasz Kaiser, Nal Kalchbrenner, Niki Parmar, Ryan Sepassi, Noam Shazeer, and Jakob Uszkoreit. 2018. Tensor2Tensor for neural machine translation. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Papers)*, pages 193–199, Boston, MA. Association for Machine Translation in the Americas.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 5998–6008.

Mingxuan Wang, Li Gong, Wenhuan Zhu, Jun Xie, and Chao Bian. 2018. Tencent neural machine translation systems for wmt18. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 522–527.

Johannes Welbl, Po-Sen Huang, Robert Stanforth, Sven Gowal, Krishnamurthy (Dj) Dvijotham, Martin Szummer, and Pushmeet Kohli. 2020. Towards verified robustness under text deletion interventions. In *International Conference on Learning Representations*.

Nianwen Xue and Yaqin Yang. 2011. Chinese sentence segmentation as comma classification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 631–635, Portland, Oregon, USA. Association for Computational Linguistics.

Zonghan Yang, Yong Cheng, Yang Liu, and Maosong Sun. 2019. Reducing word omission errors in neural machine translation: A contrastive learning approach. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6191–6196.

Lei Yu, Laurent Sartran, Wojciech Stokowiec, Wang Ling, Lingpeng Kong, Phil Blunsom, and Chris Dyer. 2020. Better document-level machine translation with Bayes' rule. *Transactions of Association for Compututational Linguistics*, 8:346–360.

In the appendix, we additionally include the description of our specialized methods for handling long sequences and the MCTS decoding algorithm. The candidate translations generated by MCTS decoding are added to the candidate pool of the noisy channel reranker in our non-primary system.

A Length Considerations

The WMT Chinese evaluation data presents documents as a sequence of segments. These segments are sentence-like units that were delimited in the original article by either unambiguous structural transitions, such as the end of a headline, or unambiguous end-of-sentence punctuation (.!?). However, in some contexts, a complete Chinese sentence may be ended with a comma (Xue and Yang, 2011). This leads to disproportionately more segments in the evaluation data being multi sentence than in the training data, which consists primarily of paired sentences, phrases, and words. Since generalization from short sequences to longer ones is a weakness of neural sequence to sequence models (Lake and Baroni, 2018), to ensure that our candidate pool contains adequate translations of long sentences, we had special handling for long sequences.

A.1 Length Analysis

There is a strong linear relationship between the number of Chinese words in the source segment and the number of English words in the translation (Figure 1 Left). The translations from our initial system were able to match this relationship when the source segment contained less than 60 words. After this point, the translations became too short. Inspection of these translations showed that the primary cause of failure was emitting the EOS token too early. Since the translations were good up to the point of truncation, we focused on methods to prevent early termination. Our final pool consisted of candidates from our original proposal models only for sentences that had less than 80 words and the rest were generated from the techniques outlined here.

A.2 Length Experts

We trained a number of length expert models with a sequence length of 384 tokens. As few ($\leq 1\%$) sentences in the training data were longer than our default sequence length, we used a mixture of real parallel data, synthetic data as described in §**??**, and concatenation of consecutive synthetic sentences to train these length experts. In addition, we also used the original proposal models which were further fine-tuned with long sequences($\geq$ 60 tokens) as additional length experts. These specialized models to handle longer sentences were used to generate proposals for sentences between 60 and 100 words.

A.3 Sentence Segmentation

We found that a lot of the long ($\geq$ 60 words) sentences in our dataset had complete sentences concatenated with commas, semicolons, full-stops, exclamations and question marks. While the latter ones are all conclusive end of sentences, commas are ambiguous as an end of sentence. Hence, we built a comma classifier that distinguished commas that signify end-of-sentence from the normal commas. While training data for this classifier was generated as outlined in (Xue and Yang, 2011), our classification model had feed-forward layers on top of the Transformer encoders (further fine-tuned on this task) that we trained for our translation task. During inference time, we recursively split every sentence on standard end-of-sentence punctuation and then semicolons followed by terminal commas (as determined by the comma classifier), into reasonably sized segments (10-60 words); Very short segments ($<$ 10 words) were merged with their neighboring segments. After this first wave of splits, if long segments ($\geq$ 60 words) still persisted, we further split them recursively on all colons, commas and reverse commas into segments between 40-60 words. Each segment was then translated independently using our translation models. This segmentation procedure was used to generate proposals for all sentences that had more than 60 space-separated words.

Sentence remerging model For each split of a sentence, we obtain a list of candidate translations, and need to combine this. For n splits with k translations each, we have k^n translations in total. Provided n and k are reasonable, we can enumerate these, but it is still too many candidates to present directly to our global reranker, so we need to have a "local" reranker that will select the best k' of these.

To select these, we define a reranking model in terms of our usual features (language model log probability for the remerged sentence to ensure coherence, channel log probabilities for the remerged candidate, sum of the direct translation log probabilities for each segment, and the total length). We

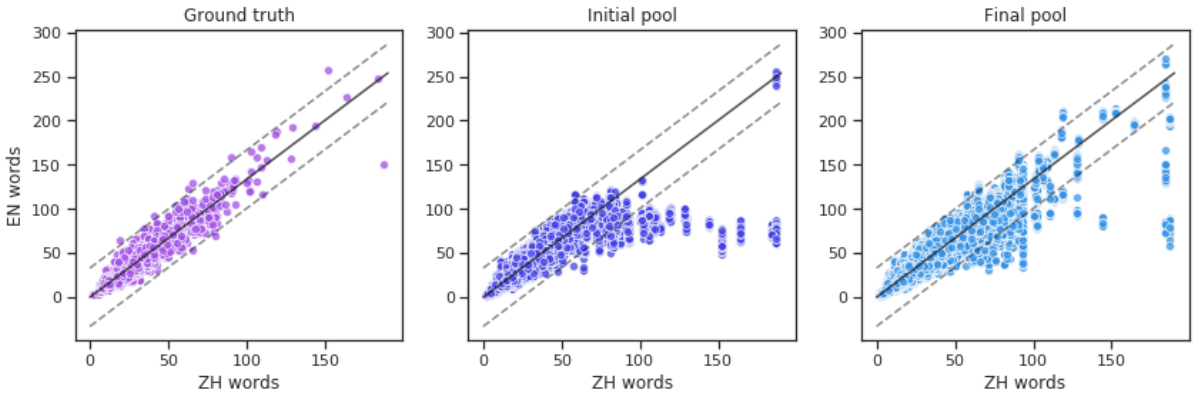


Figure 1: Relationship between the number of white space separated words on normalized text translation pairs. Left is the ground truth for our validation and test datasets. Middle is our initial candidate pool for this data while Right is our pool including candidates from length expert models and our segmentation technique. The black line is the maximum likelihood fit under $|\boldsymbol{y}_i| \sim \mathcal{N}(\beta \cdot |\boldsymbol{x}_i|, \sigma)$ with the dotted grey dashed lines representing the 3σ bounds.

select the remerged candidates k' maximizing an approximate lower bound of the corpus-BLEU.

Approximation to BLEU The weights of the features are learned so as to minimize an approximation to the expected negative log BLEU score. For the reference sentence $\boldsymbol{y}$ and hypothesis $\hat{\boldsymbol{y}}$, the negative expected log BLEU score is defined as:

$$\mathcal{L} = -\mathbb{E}\Big[\min\{0, 1 - |\boldsymbol{y}|/|\hat{\boldsymbol{y}}|\} + \sum_{i=1}^{4} \log c_i(\hat{\boldsymbol{y}}, \boldsymbol{y}) - \log r_i(\boldsymbol{y})\Big],$$

where c_i is a function that counts the clipped i-gram matches against a reference, and r_i counts the i-grams in a reference (Papineni et al., 2002).

Although our model assigns probabilities independently to sentences, BLEU is defined on an entire corpus, and because of the nonlinear functions in BLEU, we cannot compute this expectation tractably. We therefore approximate it by moving the expectations inside the nonlinear functions:

$$\mathcal{L} \approx -\Big[\min\{0, 1 - |\boldsymbol{y}|/\mathbb{E}[|\hat{\boldsymbol{y}}|]\} + \sum_{i=1}^{4} \log \mathbb{E}[c_i(\hat{\boldsymbol{y}}, \boldsymbol{y})] - \log r_i(\boldsymbol{y})\Big].$$

In this approximation, the corpus-level expectations for i-gram counts and the length can be computed tractably using the linearity of expectation. To obtain a learning algorithm, we differentiate this quantity with respect to the weightings of the scoring models and perform gradient descent.

Even with our small number of features, this objective has many local optima and in practice we run the optimizer starting from different positions and find a solution that obtains a high BLEU score and highly weights the language model (during development, we noticed that a higher weight to LM probabilities corresponded to noticeably more fluent translations, even if there was little difference in BLEU).

B MCTS Candidates

The candidate pool generated by the sequence to sequence model optimize the search space that is favorable according to those models and may fail find certain translations that score poorly according to sequence to sequence models, but receive high scores from the noisy channel model. In order to include such translations into the candidate pool, we employ Monte Carlo Tree Search (MCTS), to optimize the noisy channel objective directly. MCTS does not require a partial translation evaluation function, as opposed to Beam Search, making it an appropriate choice for decoding non-factorizable objective functions.

We define the translation environment as a progressive left-to-right language generation process, where each state $s_{\boldsymbol{y}}$ defines a sequence of tokens $\boldsymbol{y}$, and each action a_w appends a word type w at the end of the sequence generating a new state $s_{\boldsymbol{y}'}$. When an action appends the end of sentence token, a terminal state is generated. The reward R for terminal states corresponds to a log-linear combi-

nation of model scores $\phi \in M$ as follows:

$$R(s_{\boldsymbol{y}}) = \exp\left(\sum_{i \in M} \lambda_i \phi_i(\boldsymbol{y})\right), \qquad (7)$$

where λ denotes the weights attributed to each of the models. Our MCTS decoder is composed of two optimization processes running simultaneously. The former aims at finding the optimal state $R(s_{\boldsymbol{y}})$ for each of the sentences in the validation and test sets. The latter maximizes the correlation between the BLEU score on the validation set by optimizing the weights λ.

B.1 Monte Carlo Tree Search with Log Linear Models

MCTS decodes a sentence by growing a search tree. Each node in the tree corresponds to a state but adds additional statistics in order to optimally expand the search tree. Expansion is achieved by applying actions to existing nodes in the tree, generating child nodes. The search process starts with a tree with root node, which corresponds to an empty translation, and gradually expands the tree by append new words to existing nodes in the tree.

Each MCTS iteration performs the following four steps: **selection**, **expansion**, **simulation** and **backpropagation**.

The **selection** step aims at choosing the most likely node in the tree to generate the optimal translation. We employ a standard criteria UCT (Upper Confidence Bounds for Trees (Kocsis and Szepesvri, 2006)), which selects nodes recursively starting from the root according to the following criteria:

$$\text{UCT}(s) = Q(s) + b\sqrt{\frac{2 \ln N(s')}{N(s)}},$$

where s denotes the current node and s' is the parent node. $N(s)$ denotes the number of times s was traversed by the selection process and $Q(s)$ denotes the average reward obtained from s in the $N(s)$ traversals. b is a constant that quantifies the trade-off between exploitation and exploration. We set it to $b = 1$ in our experiments.

In general the average value $Q(s)$ is computed by accumulating the reward $V(s)$ obtained one any traversals containing s, then computing $Q(s) = \frac{V(s)}{N(s)}$. However, as our reward function (Eqn. 7) is concurrently updated by optimizing the weights λ, we store the accumulative individual scores $V_i(s)$ of each of the models ϕ_i. Prior to a MCTS iteration, we update λ and compute $Q(s)$ as follows:

$$Q(s) = \frac{\exp(\sum_{i \in M} \lambda_i V_i(s))}{N}.$$

This allows changes to the weights to be directly reflected in the entire search tree without the recomputation of any tree statistics. As for models M, we trained 9 proposal models and 7 channel models with the architecture defined in §4.1, and 8 language models with the architecture defined in §5.1.

Once a node s is selected, a new child s' is added to the tree in the **expansion** step.

The **simulation** step attempt to compute the expected reward for s'. This is generally accomplished by performing multiple random rollouts starting from state s', where actions are sampled from an uniform distribution until a terminal state is found. These states are then scored according to Eqn. 7. The estimate of the expected reward of s' is computed as the mean of the scores of different rollouts. However, the sparsity underlying natural language generation and the computational complexity of the scoring function makes this practice computationally challenging. Rather than performing multiple rollouts that sample from an uniform distribution at each timestamp, we perform a single rollout that runs greedy decoding starting from the translation prefix defined from state s'. As it is computationally expensive to perform decoding for each MCTS iteration, use a light-weight proposal model trained on the same data, but with a simpler architecture. For this purpose, we reduce the architecture described in §4.1 into a single layer seq2seq layer with 128 hidden units. While this network underfits the data, we found that this trade-off is desirable as it reduces the large dimensionality of the vocabulary to the few examples that are sensible at each prefix leading to a significant speed-up. The quality of the found translation remains unaltered as the reward is still computed with the full models.

Finally, each of the model scores $V_i(s')$ is propagated to all nodes from the root to s' in the **backpropagation** step.

B.2 Pairwise Reranking Optimization

In order to optimize the weights λ in Eqn. 7, we we employ the weight optimization method for log-linear models described in (Hopkins and May, 2011), which allows us to optimize our log-linear

model with respect to the non-differentiable objective function that is BLEU. This method, denominated as PRO, approximates the objective function by training a binary classifier, such that two translations $\boldsymbol{y}$ and $\boldsymbol{y}'$ respect the following equality:

$$g(\boldsymbol{y}) > g(\boldsymbol{y}') \Leftrightarrow \sum_{i \in M} \lambda_i(\phi_i(\boldsymbol{y}) - \phi_i(\boldsymbol{y}')) > 0,$$

where g is the objective function, namely BLEU. Thereby, this requires the generation of pairs $\boldsymbol{y}$ and $\boldsymbol{y}'$, where the LHS property holds. These samples are then used as data to train the model defined in the RHS.

We generate the data by sampling from the MCTS tree for each sentence pair in the development set. As $Q(s)$ is the expected score of a log-linear model with probabilities as components ϕ, we expect that $Q(s)$ is bounded in the $[0, 1]$ interval. Thus, at node s', the probability of sampling child s is given by $\frac{Q(s)}{\sum_{c \in C(s')} Q(c)}$, where $C(s)$ denotes all children of node s'. Once a leaf node is found, if it's terminal we sample its translation, otherwise we sample the translation obtained from its rollout.

Both MCTS search and PRO optimization are executed in parallel, as the former needs optimal weights in order to optimally grow the search tree, and the latter needs the search tree to generate candidates. Thus, at each iteration, the PRO optimizer samples from the most updated version of the tree for each data point in the development set, and updates the set of weights λ, which are then used in the subsequent MCTS iterations.

The NiuTrans Machine Translation Systems for WMT20

Yuhao Zhang[1], Ziyang Wang[1], Runzhe Cao[1], Binghao Wei[1], Weiqiao Shan[1], Shuhan Zhou[1], Abudurexiti Reheman[1], Tao Zhou[1], Xin Zeng[1], Laohu Wang[1], Xiaoqian Liu[1], Xunjuan Zhou[1], Yongyu Mu[1], Jingnan Zhang[1], Yinqiao Li[1], Bei Li[1], Tong Xiao[1,2] and Jingbo Zhu[1,2]

[1]NLP Lab, School of Computer Science and Engineering,
Northeastern University, Shenyang, China
[2]NiuTrans Research, Shenyang, China
yoohao.zhang@gmail.com, wangziyang@stumail.neu.edu.cn
{xiaotong,zhujingbo}@mail.neu.edu.cn

Abstract

This paper describes NiuTrans neural machine translation systems of the WMT20 news translation tasks. We participated in Japanese↔English, English→Chinese, Inuktitut→English and Tamil→English total five tasks and rank first in Japanese↔English both sides. We mainly utilized iterative back-translation, different depth and widen model architectures, iterative knowledge distillation and iterative fine-tuning. And we find that adequately widened and deepened the model simultaneously, the performance will significantly improve. Also, iterative fine-tuning strategy we implemented is effective during adapting domain. For Inuktitut→English and Tamil→English tasks, we built multilingual models separately and employed pretraining word embedding to obtain better performance.

1 Introduction

This paper describes the NiuTrans submissions to the WMT20 news tasks, including English→Chinese (EN→ZH), Tamil→English (TA→EN), Inuktitut→English (IU→EN) and Japanese↔English (JA↔EN) five directions and all of our systems were built with constrained data sets. Some useful methods in the WMT18 (Wang et al., 2018) and WMT19 (Li et al., 2019) submissions are also reused this time, such as model ensemble, knowledge distillation (KD) et al., and we explore some novel approaches this year.

For this participation, we experimented with some deeper and wider Transformer (Vaswani et al., 2017) architectures to get reliable baselines, nucleus sampling (Holtzman et al., 2020) in back-translation to generate more suitable pseudo bilingual sentences, more effectively fine-tuning strategy to adapt domain. Particularly in the low-resources tasks, {TA,IU}→EN, we built multilingual neural machine translation by using some similar language to get better performance and further replaced decoder's word embedding by an English pretraining Transformer language model's which trained by two monolingual in-domain data corpora.

Furthermore, we presented a new fine-tuning pattern which could significantly improve the BLEU score on the test set, and it worked well on all five tasks whether it is a low or rich resource. We carefully rethought this strategy and found the main gain came from domain adaptation and improved inferior translations.

Our systems and this paper followed six main steps:1) data preprocessing and filter, 2) iterative back-translation to generate pseudo bilingual data, 3) using different model architectures to enhance the diversity of translation, 4) iterative knowledge distillation by in-domain monolingual data, 5) iterative fine-tuning with in-domain using small training batch, 6) translation post-process.

2 System Overview

2.1 Data Preprocessing and Filtering

For EN→ZH and JA↔EN tasks, we first normalized the punctuation in Chinese and Japanese monolingual data by using Moses (Koehn et al., 2007) `normalize-punctuation.perl` script. English and Inuktitut sentences were segmented by Moses, while Chinese, Japanese and Tamil used NiuTrans (Xiao et al., 2012), MeCab[1] and IndicNLP[2] separately for word segmentation. After converting numbers and punctuation into English pattern, and then we normalized English words in Japanese sentences to Japanese by using Sudachi (Takaoka et al., 2018).

As previous work (Wang et al., 2018) indicated that it's important to clean data strictly, so this year

[1]https://github.com/taku910/mecab
[2]https://github.com/anoopkunchukuttan/indic_nlp_library

Proceedings of the 5th Conference on Machine Translation (WMT), pages 338–345
Online, November 19–20, 2020.

we used a stricter data filter scheme than Li et al. (2019) and the rules were following:

- Filter sentences length ratio lower than 0.4 or upper than 3 and punctuation ratio more than 0.3.
- Remove sentences that have the long word which consist of more 40 characters or words more than 200.
- Remove repeated n-gram translation and repeated sentences except for IU.
- Filter out the sentences whose alignment scores obtained by fast-align are lower than -6.
- Detecting language and delete other languages or have a special HTML label.
- Filter sentences in which parentheses on both sides do not correspond.
- Use Unicode to filter sentences that other characters more than 10.

And when we cleaned monolingual data still employed those rules and particularly there were some lines which include two or more sentences, we write a script to cut them into several sentences.

2.2 Iterative Back Translation

Back-translation is an effective way to boost translation quality by using mono data to produce pseudo training parallel data. Also, it can alleviate domain adapted problems by carefully choosing the in-domain target data. As Edunov et al. (2019); Bogoychev and Sennrich (2019) stated, due to the test target side only consisted of manual translations, back translation didn't bring evident BLEU increase on the test set. Despite our experiments proved that the deeper architectures still showed apparent improvements as the number of data increases.

As Li et al. (2019) stated, it's crucial to select in-domain mono data for back-translation. After picking out English mono data, we first used 50 million news data to train a language model (LM) built with Transformer structures, then ranked cleaned mono data which scored by trained language model before. However, it's hard to find massive in-domain data for other languages to train a neural LM, so the better choice was using a statistical method, in here we selected XenC toolkit[3] (Rousseau, 2013). The in-domain data consisted of the valid set source side and News Commentary high-quality mono data. For avoiding the short sentence ranked too high, each score was multiplied by a length penalty when using both approaches to score these data.

We chose a sample base model as our back-translation model rather ensemble model which may gain a little improvement, but needed spending huge decoding time. For multilingual model back-translation, we followed Johnson et al. (2017)'s work adding a target language label in the source side, so translations could be adapt to the target language.

This year we also followed previous work Edunov et al. (2019) and we added a new pseudo data produce methods–Nucleus Sampling, according to Holtzman et al. (2020)'s work. For all tasks we participated in, we first employed the beam search approach to generate the best translations as pseudo data and the scale of the pseudo was about 1:1 to real data. Then merge those data to retrain model and do back-translations again. Repeated those steps until the valid set BLEU have few increases then stop iterative back-translation processing. Notably, during the second back-translation, for EN→ZH task we used topk sampling and the k is 10 following last year, while for JA↔EN tasks, nucleus sampling method which the p was set 0.9 preferred better comparing topk, whereas for other tasks, {TA,IU}→EN, simply sampling was better.

2.3 Multilingual Model

For TA→EN and IU→EN, building a multilingual model is a simple and effective way to boost performance because of knowledge transfer. For TA→EN task, we added six other similar languages and only one language Russian (RU) for IU→EN task, because there were no other languages witch a relationship with IU. For TA, We up-sampled the TA data then shuffled all the train data so that each training batch could have TA data with high-probability. As for IU, we only added 0.3 million RU high quality data, then we directly merged two languages as training data. To enhance the effect of transfer learning, we utilized only one model which all the language shared the same parameters including word embeddings and vocab. Bilingual data were reused to fine-tune the model for adapting parameters to the target language after model convergence.

[3] https://github.com/antho-rousseau/XenC

Model Tag	Depth	Hidden Size	Filter Size	RPR Attention
Base	6	512	2048	✗
Big	6	1024	4096	✗
Deep25	25	512	2048	✗
Deep25-filter	25	512	4096	✗
Deep30-RPR	30	512	2048	✓
DLCL35-RPR	35	512	2048	✓
DLCL40-RPR	40	512	2048	✓
Deep15-filter-768-RPR	15	768	4096	✓

Table 1: Transformer Architectures.

2.4 Model Architectures and Ensemble

Inspired by deep network Wang et al. (2019), we tried to use simple deep, or deep and wide network architectures based on the Transformer to explore the relationship of performance and model parameters. We mainly carried out experiments on the structures of the model in Table 1. And we kept six decoder layers unchanged because it only could gain a few improvements tough many model parameters increased.

Deep Network: This model structure simply changes encoder layers, hidden size and other hyper-parameters based on vanilla Transformer.

DLCL Network: For a deeper network, we employed DLCL (Wang et al., 2019) to get more diverse models.

Filter size: This hyper-parameter represents the dimension size of feed-forward network (FFN) and simply increasing this could bring some improvements (Wang et al., 2018; Sun et al., 2019; Bawden et al., 2019). Notably, when using the deep Transformer architecture, the training time and model parameters will increase sharply with the augment of the FFN size.

RPR and relative length: The relative position representation (RPR) (Shaw et al., 2018) improves self-attention by adding relative position information. The relative length which we set 8 is the key parameter of this method.

For choosing models to ensemble, we utilized the ensemble search method which used a script to traverse all possible combinations then recorded the best one. For JA↔EN, we chose 6 of 10 while other tasks were 4 of 10.

2.5 Iterative KD and Fine-tuning

Sun et al. (2019) showed the self-learning strategy is a very effective approach to improve performance when the test set only composed of manual translations and we mainly reused (Li et al., 2019) iterative KD strategy to implement self-learning. Specifically, we designed a new iterative fine-tuning process which consists of three steps: 1)using ensemble models to decode valid and test source side sentences then fine-tune models with those pseudo data, 2) fine-tune with the valid set by a small training batch and learning rate, 3) self-learning with in-domain data which chose by only test source side. Repeat these steps two or three times according to the increase of the valid score in the third step. Figure 1 shows these steps. Notably, for being consistent with the composition of the test set, we picked out the data that the source side is real while the target side is manual from the previous valid set. In this way, we found that iterative fine-tuning can promote news title translation quality.

2.6 Reranking

For JA↔EN tasks, we followed the Ng et al. (2019), using a neural language model, and a reverse translation model. Different from the last year, we used several length penalties to generate more candidates.

2.7 Post Editing

For tasks to the English side, we only confirmed the numbers whether to generate correctly by designing a rule-based script which generated two lists for source and target sentences separately. For EN→ZH, the strategy was the same as the last year Li et al. (2019) and particularly dealt with the name's translation by using rules to delete the English name copy in Chinese sentences. For EN→JA task, we transferred English punctuation to Japanese pattern.

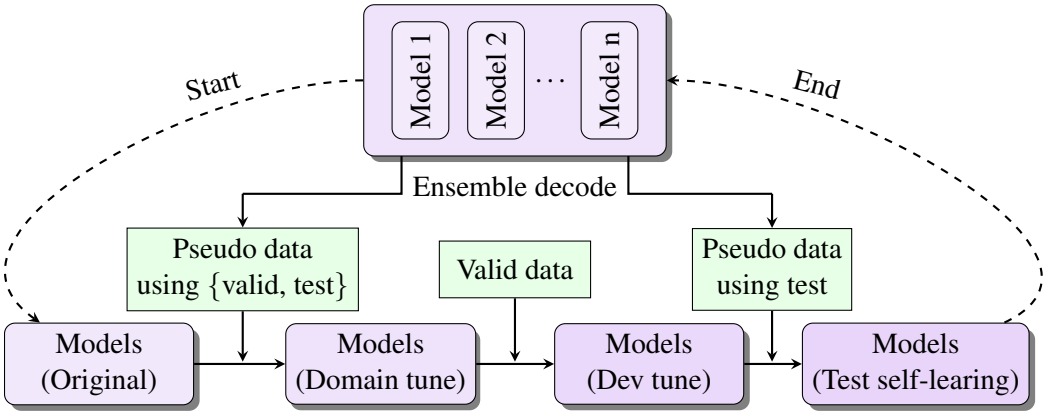


Figure 1: Iterative fine-tuning process

3 Experiment

3.1 Experiment Settings

For all tasks, we implemented the Transformer-Base as our baseline and all of our architectures were pre-normalize Wang et al. (2019) for stable training except Transformer-Big. We implemented models based on Fairseq (Ott et al., 2019) and trained on eight 2080Ti GPUs. We used Adam optimizer (Kingma and Ba, 2014) during training, $\beta_1 = 0.9$, $\beta_2 = 0.997$ for pre-normalize architectures and training batch was 2048 token while we accumulated gradient 4 times for achieving bigger batch size. We shuffled the training data before generate training batch and the training batch each epoch, so we didn't consider the document information. The max learning rate and warmup-steps we set were 0.002 and 8000 separately for deep models but 0.0016 and 16000 for deep and wide models. During training, we used fp16 to accelerate training with few performance damage. Training 15 epoch was enough for most Tasks, while 20 epoch was better for EN→ZH task and we implemented Li et al. (2020)'s methods to accelerate training. To get more robust models, the last 5 checkpoints were saved every 5000 steps for EN→ZH and JA↔EN tasks but every epoch for TA→EN,IU→EN tasks were average ensemble. During back-translation, we followed Hu et al. (2020)'s approaches to accelerate decoding when generating pseudo data.

3.2 JA↔EN Results

For JA↔EN tasks, we chose ParaCrawl v5.1, New Commentary v15, WikiMatrix, Japanese-English Subtitle Corpus, The Kyoto Free Translation Task Corpus, TED Talks total six parallel data corpus about 14.35 million and News crawl, News Commentary, Common Crawl , TED Talks 4 Japanese monolingual data corpus about 1.7 billion. After the data filter, 12 million parallel data was left and 11 million selected by the neural language model was used as training data. Cleaning several billion low-quality monolingual data will cost too much time, so here we shuffled all the data then split it into dozens of parts, one of which was 20 million. Finally we used total eight of them, each piece was carefully cleaned. Before we also used BPE (Sennrich et al., 2016) models with 32,000 merge operations for both sides to reduce UNK size in vocabulary.

We implemented back-translation two times, the first was beam search while the second was Nucleus Sampling to generate translations. Each time we selected 12 million mono data sampled from all the remaining data. Tough the second time didn't increase significantly compared with the first time, the performance was further improved with the increase of the model parameters. Considering the training time, we finally chose 35 million training data on both sides. Notably, as the official stated that the test target side only consists of manual translations, so the back-translation didn't bring too many improvements, only +0.55 and +2.1 BLEU separately in two tasks.

In order to get more diverse models for ensemble and achieve better results, we trained total 10 models including that eight with different architectures which have been shown in Table 1 and other two with different training data which consisted of 11 million bilingual data and 12 million pseudo data produced by the first back-translation. Then we searched from all the models to find the best combination of 6 out of 10 models. And Table 2 showed that the ensemble is still a robust and effective way

System	JA→EN Valid	JA→EN Test	EN→JA Valid	EN→JA Test
Baseline	19.9	20.4	33.2	34.8
+ 12M Beam	21.0	20.8	36.5	36.8
+ 12M Nucleus	21.2	21.0	36.7	36.9
Deep15-filter-768-RPR	23.2	22.9	39.1	39.3
+ Iterative KD	24.4	24.6	39.8	40.1
+ Iterative fine-tuning	25.6	26.2	40.7	41.6
+ Ensemble	25.8	26.5	41.1	41.9
+ Post Edit	26.4	26.6	42.1	42.8

Table 2: BLEU scores on JA↔EN tasks

to boost translation quality.

We implemented iterative KD process twice and each time chose 0.3 million monolingual data using ensemble model to decode then trained 3 to 5 epoch according to the dev PPL. Then we iteratively fine-tuned the models three and two times for JA→EN and EN→JA separately. And interestingly in some real case, the translation of the news titles was significantly improved after iteratively fine-tuning.

As Table 2 shows, iterative KD and fine-tuning strategies could significantly increase the BLEU on the test set.

We used the reranking model like Ng et al. (2019), though it could boost 0.3 BLEU on dev set, it didn't get benefits on the test set. During post edit, we mainly checked the number according to the source side, it also could on EN→JA task.

3.3 EN→ZH Results

In EN→ZH task, we employed News Commentary v15, UN Parallel Corpus V1.0, Back-translated news, CCMT Corpus total four corpora, and after data filter, 10 million data were sampled to train out baseline model. We set wmt18 and wmt19 test as the valid set and mainly referred wmt19 set. In the back-translation, 10 million mono data were sampled from News crawl, News Commentary and Common Crawl three corpora then used the baseline model decode by beam search strategy during the first time. During the second time, we still utilized the same amount of pseudo data while topk sampling which the k is 10 were used to translation mono sentences. From Table 3, we could find that back-translations didn't perform well. Finally 30 million data in total were used to train 10 models, different from other tasks, here we searched the best two combinations of 4 out of 10 models for iterative KD strategy to ensure the diversity of models.

System	news2019	news2020
Baseline	35.4	40.8
+ 10M Beam	36.3	41.6
+ 10M TopK	36.1	41.5
Dlcl25-RPR	38.7	44.2
+ Iterative KD	39.4	45.4
+ Iterative fine-tuning	39.8	45.9
+ Ensemble	40.1	46.7
+ Post Edit	40.3	47.3

Table 3: BLEU (%) scores on EN→ZH task

Then we implemented three times iterative KD and each time sampled 10 million in-domain source data. Table 3 showed that it's a very effective method to get 0.8 improvements. Furthermore, we fine-tuned models iteratively three times to domain adaptation and improved +0.5 BLEU. Due to implementing two ensemble combinations to decode sentences, at last model ensemble was still effective to gain 0.8 improvement. According to the WMT19 test, we adjusted the name's translations pattern during the post edit step then resulting in a 0.6 BLEU performance increase.

3.4 IU→EN Results

In IU→EN task, we only used Nunavut Hansard Inuktitut-English Parallel Corpus 3.0 total 1.3 million sentences. After the data filter, 1.1 million data was left to build the baseline model. Though romanization Inuktitut data directly was not effective, it performed better than baseline when build a multilingual system by adding 0.3 million Russia data which has the most similar semantic with Inuktitut. After that, we implemented data augmentation

System	Valid	Test
Baseline	29.6	21.3
+ Romanization	29.3	21.0
Multilingual baseline	30.2	21.6
+ 1.3M Beam	30.6	22.2
+ 1.3M Sampling	30.9	22.2
Deep15-filter-768-RPR	32.5	23.5
+ Knowledge Distillation	32.5	25.4
+ Iterative fine-tuning	49.3	28.4
+ Ensemble	49.3	28.6
+ Post Edit	49.7	29.1

Table 4: BLEU (%) scores on IU→EN task

System	Valid	Test
Baseline	12.8	13.2
Multilingual baseline	14.2	15.1
+ 0.5M Beam	19.2	15.7
+ 1M Beam	20.9	16.6
Deep15-filter-768-RPR	22.8	19.0
+ Knowledge Distillation	23.4	20.6
+ Iterative fine-tuning	23.6	20.7
+ Ensemble	23.8	21.0

Table 5: BLEU (%) scores on TA→EN task

by using the multilingual model to back-translate mono data iteratively twice and each time using 1.1 million data which equaled the true training sentences. Interestingly, the bigger and wider models improved translation quality distinctly proving it's a robust way whether the training data is rich or not.

Then we first fine-tuned the multilingual model to the target language by using 1.1 million bilingual data several epochs. According to the valid source set, ensemble models were used to decode monolingual in-domain 0.1 million data which was chosen by XenC and gained 1.85 BLEU improvement. Then fine-tuned models only once, because different from the bilingual model, the multilingual model didn't perform very well during the fine-tuning stage.

Finally we selected four models to ensemble and gained 0.18 increase, because different models were too similar after fine-tuning. And we fixed the punctuation and the score improved 0.52 BLEU. During the post process, we fixed the number and punctuation translation.

3.5 TA→EN Results

The Ta→EN task is similar to IU→EN but more complicated, because more data corpus and language can be used to build the multilingual system. Specifically, we total used {Hindi (HI), Kannada (KN), Malayalam (ML), Punjabi (PA), Telugu (TE), Urdu (UR)}→EN total six other languages, 17 million sentences according to Kudugunta et al. (2019)'s work showed similar languages with TA. From Table 5, it can be seen that using similar languages to build a multilingual system can indeed improve the performance. Also, using iterative back-translation is still an effective way but couldn't add too much pseudo language data because this will make the real target language data account for the whole data was too small, which leaded to performance damage. During the back-translation process, due to too many languages in one model, we followed Johnson et al. (2017)'s approach to build a reverse model to ensure translation quality.

For the model architectures we used, the wide and deep model was still very effective and improved 2.33 BLEU comparing with the base model. Also it performed better than simple deepen model layers. After finishing KD and fine-tuning, finally gain 1.92 improvements.

4 Conclusions

This paper introduced our submissions on WMT20 five tasks and our main exploration is using more diversified architectures, improving a iterative fine-tuning strategy and utilizing several similar languages to build a multilingual model on low-resource tasks. And we experimented with iterative back-translation by different decoding strategies, using pre-trained embeddings in multilingual models. On the whole, all of our systems performed competitively and ranked 1st on JA↔EN both sides.

Acknowledgments

This work was supported in part by the National Science Foundation of China (Nos.61876035 and 61732005) and the National Key R&D Program of China (No.2019QY1801).

References

Rachel Bawden, Nikolay Bogoychev, Ulrich Germann, Roman Grundkiewicz, Faheem Kirefu, Antonio Valerio Miceli Barone, and Alexandra Birch. 2019.

The University of Edinburgh's submissions to the WMT19 news translation task. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 103–115, Florence, Italy. Association for Computational Linguistics.

Nikolay Bogoychev and Rico Sennrich. 2019. Domain, translationese and noise in synthetic data for neural machine translation. *arXiv preprint arXiv:1911.03362*.

Sergey Edunov, Myle Ott, Marc'Aurelio Ranzato, and Michael Auli. 2019. On the evaluation of machine translation systems trained with back-translation. *arXiv preprint arXiv:1908.05204*.

Ari Holtzman, Jan Buys, Leo Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *ICLR 2020 : Eighth International Conference on Learning Representations*.

Chi Hu, Bei Li, Yinqiao Li, Ye Lin, Yanyang Li, Chenglong Wang, Tong Xiao, and Jingbo Zhu. 2020. The NiuTrans system for WNGT 2020 efficiency task. In *Proceedings of the Fourth Workshop on Neural Generation and Translation*, pages 204–210, Online. Association for Computational Linguistics.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180.

Sneha Kudugunta, Ankur Bapna, Isaac Caswell, and Orhan Firat. 2019. Investigating multilingual NMT representations at scale. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1565–1575, Hong Kong, China. Association for Computational Linguistics.

Bei Li, Yinqiao Li, Chen Xu, Ye Lin, Jiqiang Liu, Hui Liu, Ziyang Wang, Yuhao Zhang, Nuo Xu, Zeyang Wang, Kai Feng, Hexuan Chen, Tengbo Liu, Yanyang Li, Qiang Wang, Tong Xiao, and Jingbo Zhu. 2019. The niutrans machine translation systems for wmt19. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 257–266.

Bei Li, Ziyang Wang, Hui Liu, Yufan Jiang, Quan Du, Tong Xiao, Huizhen Wang, and Jingbo Zhu. 2020. Shallow-to-deep training for neural machine translation. *arXiv preprint arXiv:2010.03737*.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook fair's wmt19 news translation task submission. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 314–319.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *NAACL-HLT 2019: Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 48–53.

Anthony Rousseau. 2013. Xenc: An open-source tool for data selection in natural language processing. *The Prague Bulletin of Mathematical Linguistics*, (100):73–82.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1715–1725.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 464–468, New Orleans, Louisiana. Association for Computational Linguistics.

Meng Sun, Bojian Jiang, Hao Xiong, Zhongjun He, Hua Wu, and Haifeng Wang. 2019. Baidu neural machine translation systems for wmt19. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 374–381.

Kazuma Takaoka, Sorami Hisamoto, Noriko Kawahara, Miho Sakamoto, Yoshitaka Uchida, and Yuji Matsumoto. 2018. Sudachi: a japanese tokenizer for business. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Paris, France. European Language Resources Association (ELRA).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International*

Conference on Neural Information Processing Systems, pages 5998–6008.

Qiang Wang, Bei Li, Jiqiang Liu, Bojian Jiang, Zheyang Zhang, Yinqiao Li, Ye Lin, Tong Xiao, and Jingbo Zhu. 2018. The niutrans machine translation system for wmt18. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 528–534.

Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F. Wong, and Lidia S. Chao. 2019. Learning deep transformer models for machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1810–1822, Florence, Italy. Association for Computational Linguistics.

Tong Xiao, Jingbo Zhu, Hao Zhang, and Qiang Li. 2012. NiuTrans: An open source toolkit for phrase-based and syntax-based machine translation. In *Proceedings of the ACL 2012 System Demonstrations*, pages 19–24, Jeju Island, Korea. Association for Computational Linguistics.

Fine-grained linguistic evaluation for state-of-the-art Machine Translation

Eleftherios Avramidis, Vivien Macketanz, Ursula Strohriegel,
Aljoscha Burchardt and **Sebastian Möller**
German Research Center for Artificial Intelligence (DFKI), Berlin, Germany
firstname.lastname@dfki.de

Abstract

This paper describes a test suite submission providing detailed statistics of linguistic performance for the state-of-the-art German-English systems of the Fifth Conference of Machine Translation (WMT20). The analysis covers 107 phenomena organized in 14 categories based on about 5,500 test items, including a manual annotation effort of 45 person hours. Two systems (Tohoku and VolcanTrans) appear to have significantly better test suite accuracy than the others, although the best system of WMT20 is not significantly better than the one from WMT19 in a macro-average. Additionally, we identify some linguistic phenomena where all systems suffer (such as idioms, resultative predicates and pluperfect), but we are also able to identify particular weaknesses for individual systems (such as quotation marks, lexical ambiguity and sluicing). Most of the systems of WMT19 which submitted new versions this year show improvements.

1 Introduction

Fine-grained evaluation has recently had increasing interest on several natural language processing (NLP) tasks. Focusing on particular issues gives the possibility to analyse the automatic output in ways that cannot be seen by generic metrics. This is of particular importance in the era of deep learning, which has led to high performances and differences that are relatively difficult to distinguish. Additionally, detailed evaluation can provide indications for the improvement of the systems and the data collection, or allow focusing on phenomena of the long tail that might be of particular interest for certain cases (e.g. social biases; Stanovsky et al., 2019).

The most common method for fine-grained or focused evaluation are the *test suites* (also known as *challenge sets* or *benchmarks*; Guillou and Hardmeier, 2016; Ribeiro et al., 2020). These are test sets engineered in a particular way, so that they can test the performance of NLP tasks on concrete issues (Müller et al., 2018; Bawden et al., 2018).

This paper is presenting the use of such a test suite for the evaluation of the 11 German→English Machine Translation (MT) systems that participated at the Shared Task of the Fifth Conference of Machine Translation (WMT20; Barrault et al., 2020). The evaluation applies the DFKI test suite on German-English, via 5,514 test items which cover 107 linguistically motivated phenomena organized in 14 categories. After a reference in related work (Section 2), we explain shortly the structure of the test suite (Section 3) and present the results (Section 4) and the conclusions (Section 5).

2 Related Work

The use of test suites was introduced along with the early steps of MT in the 1990's (King and Falkedal, 1990; Way, 1991; Heid and Hildenbrand, 1991). With the emergence of deep learning, recent works re-introduced test suites that focus on the evaluation of particular linguistic phenomena (e.g. pronoun translation; Guillou and Hardmeier, 2016) or more generic test suites that aim at comparing different MT technologies (Isabelle et al., 2017; Burchardt et al., 2017) and Quality Estimation methods (Avramidis et al., 2018). The test suite track of the Conference of Machine Translation has already taken place two years in a row, allowing the presentation of several test suites, focusing on various linguistic phenomena and supporting different language directions. These include work in grammatical contrasts (Cinkova and Bojar, 2018), discourse (Bojar et al., 2018), morphology (Burlot et al., 2018), pronouns (Guillou et al., 2018) and word sense disambiguation (Rios et al., 2018). When compared to the vast majority of the previous test suites, the one presented here is the only one

Proceedings of the 5th Conference on Machine Translation (WMT), pages 346–356
Online, November 19–20, 2020.

Lexical Ambiguity	
Er las gerne Novellen.	
He liked to read novels.	fail
He liked to read novellas.	pass
Phrasal verb	
Warum starben die Dinosaurier aus?	
Why did the dinosaurs die?	fail
Why did the dinosaurs die out?	pass
Why did the dinosaurs become extinct?	pass
Ditransitive Perfect	
Ich habe Tim einen Kuchen gebacken.	
I have baked a cake.	fail
I baked Tim a cake.	pass

Table 1: Examples of passing and failing MT outputs

that performs a systematic evaluation of more than one hundred phenomena on the state-of-the-art systems participating in WMT20.

3 Method

The test suite is a test set that has been devised manually with the aim to allow testing the MT output for several linguistic phenomena. The entire test suite consists of subsets that test one particular phenomenon each, through several test items. Each test item of the test suite consists of a source sentence and a set of correct and/or incorrect MT outputs. At the evaluation time, the test items are given as input to the MT systems and it is tested on whether the respective MT output consists a correct translation. By observing the amount of the test items that are translated correctly, one can calculate the performance of the MT systems regarding the respective phenomenon.

The evaluation presented in this paper is based on the DFKI Test Suite for MT on German to English, which has been presented in Burchardt et al. (2017) and applied extensively in the WMT shared task of 2018 (Macketanz et al., 2018) and 2019 (Avramidis et al., 2019). The current version includes 5,560 test items in order to control 107 phenomena organised in 14 categories. Some sample test items can be seen in Table 1 whereas a more detailed list of test sentences with correct and incorrect translations can be found on GitHub[1].

3.1 Application of the test suite

The construction of the test suite has been thoroughly explained in the papers from the previous years (Avramidis et al., 2018, 2019) and depicted in Figure 1 (steps *a*-*c*). The test items of the test suite are given as input to the MT systems (step *d*). Their MT outputs are tested using a set of rules (regular expressions or fixed strings), each rule specific for a phenomenon, that defines whether the translations are correct with respect to the tested phenomenon (step *e*). When the automatic application of the rules cannot lead to a clear decision on whether the translation is correct or not, the test item is left with a warning. The warnings are consequently resolved by human annotators with linguistic knowledge, who inspect the MT output, provide a clear judgment and also augment the set of the rules to cover similar cases in the future (step *e*).

For every system we calculate the phenomenon-specific translation accuracy as the the number of the test sentences for the phenomenon which were translated properly, divided by the number of all test sentences for this phenomenon:

$$\text{accuracy} = \frac{\text{correct translations}}{\text{sum of test items}}$$

Each phenomenon is covered by at least 20 test items[2], whereas the same test items are given to multiple systems to achieve comparisons among them. In order to achieve a fair comparison among the systems, only the test items that do not contain any warnings for any of the systems are included in the calculation.

In order to define which systems have the best performance for a particular phenomenon, all systems are compared with the system with the highest accuracy. When comparing the highest scoring system with the rest, the significance of the comparison is confirmed with a one-tailed Z-test with $\alpha = 0.95$. The systems whose difference with the best system is not significant are considered to be in the first performance cluster and indicated with boldface in the tables.

3.2 Experiment setup

In the evaluation presented in the paper, MT outputs are obtained from the 11 systems that are part of the *news translation task* of the Fifth Conference on Machine Translation (WMT20). These are 6 systems submitted by the shared task participants, one baseline system from the shared task of the biomedical domain (WMTBiomedBaseline; Bawden et al., 2020) and 4 online commercial systems whose output has been obtained by the workshop organizers and therefore have been anonymized

[1] https://github.com/DFKI-NLP/TQ_AutoTest

[2] with the exception of 7 phenomena which have 9-19 items

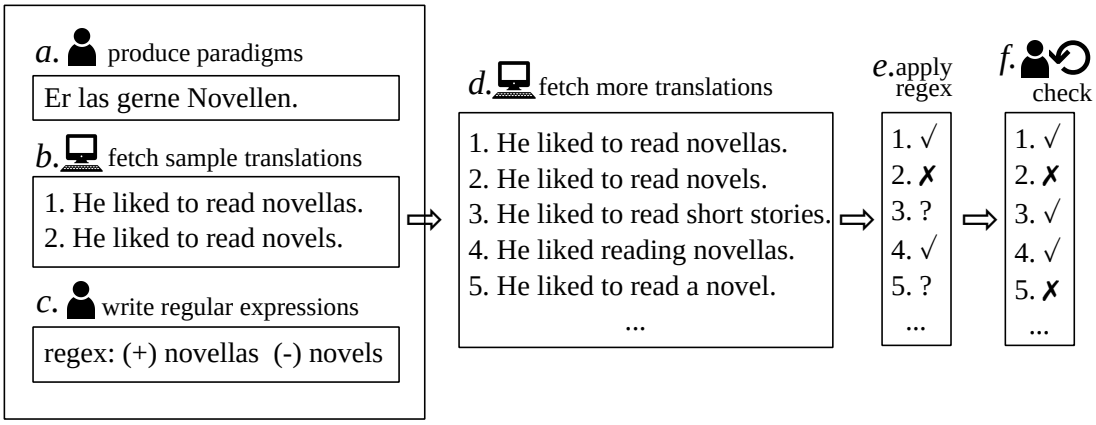


Figure 1: Example of the preparation and application of the test suite for one test sentence

(Online-A, B, G and Z). The submitted systems are OPPO (Shi et al., 2020), PROMT (Molchanov, 2020), Tohoku-AIP-NTT (Kiyono et al., 2020), UEdin (Germann, 2020), VolcTrans (Wu et al., 2020) and Zlabs, whereas a 7th system under the name "yolo"was ignored because it contained arbitrary translations. Unfortunately, contrary to previous years, very few system descriptions were provided by the time this paper was written and it is therefore not possible to associate linguistic performance with system types and settings.

As explained earlier, the application of the test suite on the output of the 11 systems left about 10% of unresolved warnings which needed to be manually edited. A human annotator with linguistic background devoted about 45 working hours in order to resolve 99% of them (resulting into 5,514 valid out of 5,560 total items).

4 Results

The accuracy of each system per linguistic category is briefly shown in Table 5 whereas the detailed statistics depicting the accuracy for every linguistic phenomenon, grouped in the respective linguistic categories are shown in Table 7. Since every category and every phenomenon have a different amount of test items, the average scores, shown in the last rows of the tables, are computed in three different ways: The first aggregates the contributions of all test items to compute the average percentages (micro-average), the second (Table 5) computes the percentages independently for each category and then takes the average (hence treating all categories equally; *category macro-average*) and the third (Table 7) computes the percentages independently for each phenomenon and then takes the average (hence treating all phenomena equally; *phenomenon macro-average*). The significantly best systems for every category or phenomenon are bold-faced.

Very high scores do not necessarily mean that the MT of the respective grammatical phenomenon has been solved, but rather that the current test items of the test suite (which was engineered with the emergence of the first neural MT systems in 2017) are unable to expose difficulties of the systems. The artificial nature of the test suite and the variable number of test items per category and phenomenon should also be taken in consideration when doing comparisons between categories and phenomena.

4.1 Comparison between systems

Two systems are standing out for their overall performance. **Tohoku** achieves the best category macro-averaged accuracy of 88.1%, whereas it is sharing the first position with **VolcTrans** based on their micro-averaged accuracy (85.3-85.4%). The systems UEdin, Online-B, Online-G and Online-A are next. Tohoku and VolcTrans are also the best performing systems for all linguistic categories, whereas UEdin is losing in one category and Online-A is losing in two categories.

Two systems, WMTBiomedBaseline and ZLabs show very low performances and are assumed to be non state-of-the-art systems. We will therefore exclude these two systems from the discussion and conclusions for phenomena and categories. Whereas no description was available for ZLabs, the lower performance of the WMTBiomedBaseline can be attributed to the fact that it was trained with only 56% of the parallel training data used by Tohoku and no synthetic data. Among the rest of the systems, the worst performing one is Online-Z, achieving the lowest accuracy (74% on both micro- and macro-average), being on par with the best systems

Verb Valency	
Ich erinnere mich seiner.	
I remember his.	fail
I remember him.	pass
False Friends	
Er las gerne Novellen.	
He liked to read novels.	fail
He liked to read novellas.	pass

Table 2: Examples of linguistic categories with lower accuracy with passing and failing MT outputs

on only 6 categories.

BLEU scores (Papineni et al., 2002) on the official test-set are also calculated for further comparison. The order of the systems based on BLEU seems to correlate with the order given by the category macro-average with the exception of one system (OPPO). Since BLEU scores are calculated on a different test set, further investigation is needed to confirm if this correlation is of any significance. According to the official human evaluation campaign (Barrault et al., 2020, table 11), the first nine systems are tied, so it is hard to compare this system order with theirs.

4.2 Linguistic categories

The average accuracy regarding the linguistic categories ranges in relatively high numbers, between 68.9% and 97.3%. The categories with the highest accuracy in average are the **negation** (97.3%), the **composition** (85.3%), the **subordination** (85.3%) and the **named entinties and terminology** (82%). The ones with the lowest accuracy are the **multiword expressions** (MWE), the **ambiguity**, the **false friends** and the **verb valency** (68.9-71.5%).

When one tries to identify weaknesses of particular systems, OPPO is suffering mostly concerning function words and long distance dependencies (LDD) / interrogatives. Online-Z and PROMT have issues with ambiguity and several systems have issues with punctuation. Some of these issues are discussed in a more fine-grained level below.

The comparison of the state-of-the-art systems with the low-resource WMTBiomedBaseline indicates that some categories, such as ambiguity and composition, are particularly sensitive to low resources, as their accuracy is proportionally lower than other categories if compared to the respective category accuracies of the state-of-the-art systems.

Table 2 contains examples from the two low accuracy categories *verb valency* and *false friends*. *Verb valency* refers to the arguments that are being controlled by the predicate. Certain verbs require a specific grammatical case. In our example, the German verb *sich erinnern* (to remember) requires a genitive object, in this case *seiner*. *Seiner*, however, can also mean *his* as in the possessive pronoun, which explains the mistranslation of *I remember his*.

False friends are words in different languages that look similar and are therefore often mistaken for being translations of one another, even though their meanings differ. The German noun *Novelle* does not translate to *novel*, but to *novella* or *short story*. While you would expect a human to make these kind of translation errors, it is surprising to see that also MT systems are prone to mistranslating false friends.

4.3 Linguistic phenomena

The accuracy regarding individual linguistic phenomena has a wide range, between very low scores (15%) and full success (100%). The phenomena which all systems had difficulty to handle were the **idioms** and the **resultative predicates**, with most systems scoring only 20% and 26% respectively. However, the overall performance on these phenomena has improved: last year only 3 systems could achieve this performance, with the majority of the systems having 5-10% less accuracy. **Modal pluperfect** is also ranging very low, scoring between 2.2% and 50.6% and similar is the case for its negated version. Other moods of the pluperfect make it particularly difficult for some systems, e.g. PROMT and Online-Z suffer in translating the **ditransitive** and **intransitive pluperfect**.

When trying to find the cases that consist a weakness for particular systems, Online-Z indicates one of the lowest scores in punctuation, which appears to derive from the fact that the system removes all **quotation marks**. A similar issue is observed with Online-G and OPPO which could correctly convey almost half of the quotation marks, whereas another two systems have some way to go. Interestingly enough, despite strongly depending on preprocessing, quotation marks are a common issue, since similar cases have been noted in previous years. In other phenomena, Online-Z has a very low accuracy for **sluicing** whereas PROMT is relatively weak concerning **lexical ambiguity**.

Table 3 contains further translation examples from linguistic phenomena with low accuracy. A *resultative predicate* is a construction that consists

Resultative Predicate	
Sie trinkt die Tasse leer.	
She drinks the cup empty.	fail
She empties the cup.	*pass*
She is drinking the whole cup.	*pass*
Intransitive Pluperfect	
Sie hatten geschlafen.	
They were sleeping.	fail
They had slept.	pass
Sluicing	
John mag die Nudeln nicht, aber er weiß nicht, warum.	
John doesn't like the noodles but he doesn't know why.	fail
John doesn't like the noodles, but he doesn't know why.	pass
Lexical Ambiguity	
Das Gericht gestern Abend war lecker.	
The court last night was delicious.	fail
The dish last night was delicious.	pass

Table 3: Examples of linguistic phenomena with low accuracy with passing and failing MT outputs

of a verb and an adjective in which the verb describes an action and the adjective describes the result of that action. In many cases, resultative predicates lead to translation errors as they do not exist in English and a literal translation leads to an ungrammatical translation, as can be seen in the example. Since none of the systems could produce a correct output for this sentence, we have provided two possible correct translations here as examples.

Intransitive verbs do not require further objects (as opposed to transitive or ditransitive verbs). *Pluperfect* is a tense which is used in German to describe completed actions that have taken place in the past. It should be translated to English in pluperfect as well. In the example, the incorrect translation contains past progressive *were sleeping* instead of the correct *had slept*.

Sluicing is a type of ellipsis that can occur in direct and indirect interrogative clauses. A wh-word precedes the part of the sentence that contains the ellipsis. In our example, all constituents following the wh-word are elided: *John mag die Nudeln nicht, aber er weiß nicht, warum ~~er die Nudeln nicht mag~~*. Sluicing exists in both German and in English: *John doesn't like the noodles, but he doesn't know why ~~he doesn't like the noodles~~*. One difference between the German and the English sluicing sentence is that in German there are two commas, while in English there is only one. Since this phenomenon concerns the complete sentence, punctuation should be correct when translating a sentence containing sluicing. In our example, the missing comma leads to fail.

The fourth example in the Table contains the lexical ambiguity *Gericht*. *Gericht* can either mean *court* or *dish* but the context provided in the sentence (*war lecker*, English: *was delicious*) serves as disambiguation so that only a translation referring to *dish* (or to food in some way) can be a pass. Any translation referring to *court/courthouse/tribunal* or the like is a fail.

WMT	categ. macro-	micro-
2018	81.0	**84.1**
2019	**87.4**	83.0
2020	**88.0**	**85.1**

Table 4: The accuracy (%) of the best system of each year as measured over 5,555 test items that were common over the last years. The scores that are significantly higher in each column are boldfaced

4.4 Comparison with previous years

One can notice some improvements on the overall performance of the best system, as compared with the previous two years. As seen in Table 4 from 2018 to 2019 there was a 6.4% improvement on the macro-averaged accuracy but there was no significant improvement from 2019 to 2020. The best system of 2019 was not submitted in 2020 and this is unfortunate, as it performed better than this year's best system in four categories (mostly regarding ambiguity; Table 6). When considering micro-averaged accuracy, there is significant improvement since last year (2.1%), but the best system of 2018 is competing with the one of this year, due to its high performance regarding verb tense/mood.

One can also consider the improvement of individual systems submitted to WMT from one year to another, starting from 2018. This year, only 5 of the 2019 systems submitted their new version and the yearly difference of their test suite accuracy can be seen in Table 6. The accuracy is measured over the test items that are common over all three years (or at least the last two).

All systems indicate considerable improvements on the macro-average since the previous year, ranging between 2,4% and 8,5%. Online-G had a major improvement for a second year in a row (21.6% in total), whereas it is the only system that achieved such an improvement without having an accuracy drop for any of the linguistic categories, whereas it improved 6 categories for more than 10%. PROMT

had improvement in all categories apart from verb tense/aspect/mood, UEdin deteriorated in verb tense/aspect/mood, whereas Online-B deteriorated in composition, named entity/terminology and punctuation.

The linguistic categories that improved mostly in average are the **long distance dependencies / interrogatives**, the **verb valency**, the **ambiguity** and the **punctuation**.

5 Conclusions and further work

In this paper we present the results of the application of the DFKI test suite in the output of the state-of-the-art MT systems participating in the Shared Task of the Fifth Conference of Machine Translation (WMT20). Based on about 5,500 test items, we present detailed accuracies regarding 107 phenomena organized in 14 categories. Additionally, the evolution of systems submitted also in previous years is observed.

The best system of this year is not significantly better than the one from 2019 in a macro-average, but one can see significant improvement from two years ago. The systems that seem to have the best accuracies are Tohoku and VolcanTrans. The phenomena that most systems face difficulties are again this year the idioms, the resultative predicates and some moods of the pluperfect, whereas some systems still have issues with quotation marks and lexical ambiguity.

As discussed previously, the high accuracies achieved for particular phenomena or categories raise questions on whether these phenomena are getting solved, or whether the test suite (which was originally built to challenge the systems from 2017) should raise the difficulty by including more test items.

In further work, we would like to be able to associate the performance on specific phenomena with decisions related to decisions during the development of the systems, once there is enough information about this process for all systems.

Acknowledgments

This research was supported by the German Research Foundation through the project TextQ and by the German Federal Ministry of Education through the project SocialWear.

References

Eleftherios Avramidis, Vivien Macketanz, Arle Lommel, and Hans Uszkoreit. 2018. Fine-grained evaluation of quality estimation for machine translation based on a linguistically motivated test suite. In *Proceedings of the AMTA 2018 Workshop on Translation Quality Estimation and Automatic Post-Editing*, pages 243–248, Boston, MA. Association for Machine Translation in the Americas.

Eleftherios Avramidis, Vivien Macketanz, Ursula Strohriegel, and Hans Uszkoreit. 2019. Linguistic evaluation of German-English machine translation using a test suite. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 445–454, Florence, Italy. Association for Computational Linguistics.

Loïc Barrault, Magdalena Biesialska, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Matthias Huck, Eric Joanis, Tom Kocmi, Philipp Koehn, Chi-kiu Lo, Nikola Ljubešić, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Santanu Pal, Matt Post, and Marcos Zampieri. 2020. Findings of the 2020 Conference on Machine Translation (WMT20). In *Proceedings of the Fifth Conference on Machine Translation*, pages 1–54, Online. Association for Computational Linguistics.

Rachel Bawden, Giorgio Maria Di Nunzio, Cristian Grozea, Inigo Jauregi Unanue, Antonio Jimeno Yepes, Nancy Mah, David Martinez, Aurélie Névéol, Mariana Neves, Maite Oronoz, Olatz Perez-de Viñaspre, Massimo Piccardi, Roland Roller, Amy Siu, Philippe Thomas, Federica Vezzani, Maika Vicente Navarro, Dina Wiemann, and Lana Yeganova. 2020. Findings of the WMT 2020 biomedical translation shared task: Basque, italian and russian as new additional languages. In *Proceedings of the Fifth Conference on Machine Translation*, pages 658–685, Online. Association for Computational Linguistics.

Rachel Bawden, Rico Sennrich, Alexandra Birch, and Barry Haddow. 2018. Evaluating discourse phenomena in neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1304–1313, New Orleans, Louisiana. Association for Computational Linguistics.

Ondřej Bojar, Jiří Mírovský, Kateřina Rysová, and Magdaléna Rysová. 2018. EvalD Reference-Less Discourse Evaluation for WMT18. In *Proceedings of the Third Conference on Machine Translation*, pages 545–549, Belgium, Brussels. Association for Computational Linguistics.

Aljoscha Burchardt, Vivien Macketanz, Jon Dehdari, Georg Heigold, Jan-Thorsten Peter, and Philip Williams. 2017. A Linguistic Evaluation of Rule-Based, Phrase-Based, and Neural MT Engines. *The Prague Bulletin of Mathematical Linguistics*, 108:159–170.

Franck Burlot, Yves Scherrer, Vinit Ravishankar, Ondřej Bojar, Stig-Arne Grönroos, Maarit Koponen, Tommi Nieminen, and François Yvon. 2018. The WMT'18 Morpheval test suites for English-Czech, English-German, English-Finnish and Turkish-English. In *Proceedings of the Third Conference on Machine Translation*, pages 550–564, Belgium, Brussels. Association for Computational Linguistics.

Silvie Cinkova and Ondřej Bojar. 2018. Testsuite on Czech–English Grammatical Contrasts. In *Proceedings of the Third Conference on Machine Translation*, pages 565–575, Belgium, Brussels. Association for Computational Linguistics.

Ulrich Germann. 2020. The University of Edinburgh's submission to the German-to-English and English-to-German Tracks in the WMT 2020 News Translation and Zero-shot Translation Robustness Tasks. In *Proceedings of the Fifth Conference on Machine Translation*, pages 196–200, Online. Association for Computational Linguistics.

Liane Guillou and Christian Hardmeier. 2016. PROTEST: A Test Suite for Evaluating Pronouns in Machine Translation. *Tenth International Conference on Language Resources and Evaluation (LREC 2016)*.

Liane Guillou, Christian Hardmeier, Ekaterina Lapshinova-Koltunski, and Sharid Loáiciga. 2018. A Pronoun Test Suite Evaluation of the English–German MT Systems at WMT 2018. In *Proceedings of the Third Conference on Machine Translation*, pages 576–583, Belgium, Brussels. Association for Computational Linguistics.

Ulrich Heid and Elke Hildenbrand. 1991. Some practical experience with the use of test suites for the evaluation of SYSTRAN. In *the Proceedings of the Evaluators' Forum, Les Rasses*. Citeseer.

Pierre Isabelle, Colin Cherry, and George Foster. 2017. A Challenge Set Approach to Evaluating Machine Translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2486–2496, Copenhagen, Denmark. Association for Computational Linguistics.

Margaret King and Kirsten Falkedal. 1990. Using test suites in evaluation of machine translation systems. In *Proceedings of the 13th conference on Computational Linguistics*, volume 2, pages 211–216, Morristown, NJ, USA. Association for Computational Linguistics.

Shun Kiyono, Takumi Ito, Ryuto Konno, Makoto Morishita, and Jun Suzuki. 2020. Tohoku-aip-ntt at wmt 2020 news translation task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 144–154, Online. Association for Computational Linguistics.

Vivien Macketanz, Eleftherios Avramidis, Aljoscha Burchardt, and Hans Uszkoreit. 2018. Fine-grained evaluation of German-English Machine Translation based on a Test Suite. In *Proceedings of the Third Conference on Machine Translation (WMT18)*, Brussels, Belgium. Association for Computational Linguistics.

Alexander Molchanov. 2020. PROMT Systems for WMT 2020 Shared News Translation Task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 247–252, Online. Association for Computational Linguistics.

Mathias Müller, Annette Rios, Elena Voita, and Rico Sennrich. 2018. A large-scale test set for the evaluation of context-aware pronoun translation in neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 61–72, Brussels, Belgium. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online. Association for Computational Linguistics.

Annette Rios, Mathias Müller, and Rico Sennrich. 2018. The Word Sense Disambiguation Test Suite at WMT18. In *Proceedings of the Third Conference on Machine Translation*, pages 594–602, Belgium, Brussels. Association for Computational Linguistics.

Tingxun Shi, Shiyu Zhao, Xiaopu Li, Xiaoxue Wang, Qian Zhang, Di Ai, Dawei Dang, Xue Zhengshan, and JIE HAO. 2020. OPPO's machine translation systems for WMT20. In *Proceedings of the Fifth Conference on Machine Translation*, pages 281–291, Online. Association for Computational Linguistics.

Gabriel Stanovsky, Noah A. Smith, and Luke Zettlemoyer. 2019. Evaluating gender bias in machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1679–1684, Florence, Italy. Association for Computational Linguistics.

Andrew Way. 1991. Developer-Oriented Evaluation of MT Systems. In *Proceedings of the Evaluators' Forum*, pages 237–244, Les Rasses, Vaud, Switzerland. ISSCO.

Liwei Wu, Xiao Pan, Zehui Lin, Yaoming ZHU, Mingxuan Wang, and Lei Li. 2020. The Volctrans Machine Translation System for WMT20. In *Proceedings of the Fifth Conference on Machine Translation*, pages 304–310, Online. Association for Computational Linguistics.

category	items	Tohoku	VolcTrans	UEdin	Onl-B	Onl-G	Onl-A	PROMT	OPPO	Onl-Z	ZLabs	WMTBi	avg
Ambiguity	81	**82.7**	**77.8**	**72.8**	**79.0**	**84.0**	**76.5**	64.2	**82.7**	67.9	45.7	30.9	69.5
Composition	49	**98.0**	**98.0**	**93.9**	**93.9**	**95.9**	**93.9**	**89.8**	**95.9**	85.7	49.0	44.9	85.3
Coordination & ellipsis	78	**89.7**	**91.0**	**89.7**	**91.0**	**85.9**	**87.2**	**87.2**	**87.2**	60.3	52.6	44.9	78.8
False friends	36	**72.2**	**80.6**	**72.2**	**80.6**	**77.8**	**69.4**	**72.2**	66.7	**86.1**	52.8	50.0	71.0
Function word	72	**86.1**	**80.6**	**86.1**	**90.3**	**90.3**	**83.3**	**88.9**	55.6	**88.9**	41.7	43.1	75.9
LDD & interrogatives	174	**89.1**	**86.2**	**85.1**	**83.3**	**86.8**	77.6	81.0	58.6	72.4	48.3	58.6	75.2
MWE	80	**80.0**	**75.0**	**71.3**	**77.5**	**77.5**	**71.3**	**70.0**	**78.8**	**73.8**	45.0	37.5	68.9
Named entitiy & terminology	89	**92.1**	**84.3**	**87.6**	82.0	82.0	**88.8**	**87.6**	**85.4**	68.5	70.8	73.0	82.0
Negation	20	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**95.0**	**100.0**	**100.0**	**95.0**	80.0	**100.0**	97.3
Non-verbal agreement	61	**91.8**	**88.5**	**88.5**	**86.9**	**90.2**	**83.6**	**82.0**	**88.5**	**85.2**	54.1	57.4	81.5
Punctuation	60	**96.7**	**98.3**	**98.3**	71.7	61.7	**100.0**	**98.3**	70.0	28.3	68.3	55.0	77.0
Subordination	180	**90.6**	**88.3**	**91.1**	**91.1**	**92.2**	**88.9**	**90.0**	**90.6**	**87.8**	65.0	62.2	85.3
Verb tense/aspect/mood	4447	**84.6**	**85.3**	80.3	75.9	79.6	77.5	75.1	79.3	73.6	50.5	52.1	74.0
Verb valency	87	**79.3**	**81.6**	**77.0**	**81.6**	**77.0**	**77.0**	**71.3**	**80.5**	64.4	44.8	51.7	71.5
micro-average	5514	**85.3**	**85.4**	81.2	77.7	80.6	78.7	76.5	79.1	73.6	51.3	52.4	74.7
macro-average	5514	**88.1**	86.8	85.3	84.6	84.3	83.6	82.7	80.0	74.1	54.9	54.4	78.1
BLEU		43.8	43.5	42.3	41.9	41.4	40.4	39.6	43.2	35.4	31.5	32.1	35.4

Table 5: Accuracies (%) of successful translations for 11 systems and 14 categories. Boldface indicates the significantly best performing systems in each row

	Onl-A		Onl-B		Onl-G		PROMT	UEdin		best	
category	2019	2020	2019	2020	2019	2020	2020	2019	2020	2019	2020
Ambiguity	+2.6	+7.7	+1.3	+2.6	+2.6	+11.5	+16.7	+11.6	+14.1	+12.4	-9.9
Composition	+10.4	+2.1		-4.1	+12.5	+12.5	+10.4	+8.3	+8.4	+12.3	
Coordination & ellipsis				+2.2	+8.7	+19.6	+6.6		+4.4	5.1	-1.3
False friends		-2.8	+2.8	+2.8		+5.6		+13.9	+5.5	+8.3	-2.8
Function word	+1.6	-1.6		+10.9	+42.2	+1.6	+6.2	+7.8	+2.7		
LDD & interrogatives	-2.2	+8.7		+6.6	+14.2	+18.4	+8.7		+15.2	+1.8	+3.4
MWE	+1.3	+5.4		+6.6	+5.3	+9.3	+14.6	+5.3	+10.7	+10.0	+1.2
Named entitiy & terminology		+3.1		-3.2	-1.5	+7.8	+15.6	+7.8	+3.1	-2.3	+9.0
Negation		-5.0		+5.0	+4		+10.0	-10.0	+10.0		
Non-verbal agreement	+6.9				+22.4	+10.4	+10.4	+10.4	+6.9	+9.8	
Punctuation	-21.9	+25.5		-3.6	-7.3	+3.7	+16.4	+21.8	+9.1	+30.0	+3.4
Subordination	-11.3	+10.3	+1.0	+7.2	+6.2	+5.1	+4.1	-2.1	+7.2	+3.8	+1.2
Verb tense/aspect/mood	+11.9	-6.3	+0.2	+2.0	+19.8	+13.9	-0.2	+5.6	-1.1	0.8	5.8
Verb valency	+1.5	+9.0		+13.5	+11.9	+6.0	+11.9	+3.0	+13.4	9.2	-1.2
micro-avg	+9.9	-4.4	+0.2	+2.4	+18.5	+13.1	+1.5	+5.6	+0.6	+2.0	+4.8
macro-avg	+0.1	+4.0	+0.4	+3.5	+12.7	+8.9	+9.3	+6.0	+7.6	+7.4	+0.6

Table 6: Difference of the test suite accuracy from one year to the next one per category, for the systems participating in the shared tasks WMT18-20, measured over the test items that are common over all these years.

phenomenon	items	Tohoku	VolcTrans	UEdin	Onl-B	Onl-G	Onl-A	PROMT	OPPO	Onl-Z	ZLabs	WMTBi	avg
Ambiguity	81	**82.7**	**77.8**	**72.8**	**79.0**	**84.0**	**76.5**	64.2	**82.7**	67.9	45.7	30.9	69.5
Lexical ambiguity	63	**85.7**	**79.4**	**77.8**	**77.8**	**87.3**	**79.4**	65.1	**85.7**	68.3	49.2	36.5	72.0
Structural ambiguity	18	**72.2**	**72.2**	**55.6**	**83.3**	**72.2**	**66.7**	**61.1**	**72.2**	**66.7**	33.3	11.1	60.6
Composition	49	**98.0**	**98.0**	**93.9**	**93.9**	**95.9**	**93.9**	**89.8**	**95.9**	85.7	49.0	44.9	85.3
Compound	29	**96.6**	**96.6**	**89.7**	**93.1**	**93.1**	**89.7**	**86.2**	**96.6**	**82.8**	51.7	48.3	84.0
Phrasal verb	20	**100.0**	**100.0**	**100.0**	**95.0**	**100.0**	**100.0**	**95.0**	**95.0**	**90.0**	45.0	40.0	87.3
Coordination & ellipsis	78	**89.7**	**91.0**	**89.7**	**91.0**	**85.9**	**87.2**	**87.2**	**87.2**	60.3	52.6	44.9	78.8
Gapping	20	**95.0**	**95.0**	**95.0**	**95.0**	**95.0**	**80.0**	**95.0**	**90.0**	**75.0**	70.0	40.0	84.1
Right node raising	20	**80.0**	**85.0**	**85.0**	**80.0**	**65.0**	**85.0**	**80.0**	**85.0**	**85.0**	40.0	15.0	71.4
Sluicing	18	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	11.1	72.2	**100.0**	89.4
Stripping	20	**85.0**	**85.0**	**80.0**	**90.0**	**85.0**	**85.0**	**75.0**	**75.0**	**65.0**	30.0	30.0	71.4
False friends	36	**72.2**	**80.6**	**72.2**	**80.6**	**77.8**	**69.4**	**72.2**	66.7	**86.1**	52.8	50.0	71.0
Function word	72	**86.1**	**80.6**	**86.1**	**90.3**	**90.3**	**83.3**	**88.9**	55.6	**88.9**	41.7	43.1	75.9
Focus particle	24	**91.7**	**91.7**	**95.8**	**95.8**	**100.0**	**91.7**	**95.8**	**91.7**	**95.8**	75.0	83.3	91.7
Modal particle	29	**75.9**	**72.4**	**72.4**	**79.3**	**75.9**	**72.4**	**79.3**	**62.1**	**79.3**	41.4	37.9	68.0
Question tag	19	**94.7**	78.9	**94.7**	**100.0**	**100.0**	**89.5**	**94.7**	0.0	**94.7**	0.0	0.0	67.9
LDD & interrogatives	174	**89.1**	**86.2**	**85.1**	**83.3**	**86.8**	77.6	81.0	58.6	72.4	48.3	58.6	75.2
Extended adjective construction	20	**85.0**	**80.0**	**75.0**	**80.0**	**80.0**	**70.0**	**75.0**	**80.0**	55.0	55.0	50.0	71.4
Extraposition	20	75.0	70.0	75.0	55.0	65.0	65.0	75.0	65.0	65.0	40.0	40.0	62.7
Multiple connectors	20	**90.0**	**90.0**	**95.0**	**75.0**	**95.0**	**75.0**	70.0	**80.0**	65.0	**85.0**	**85.0**	82.3
Pied-piping	20	**90.0**	**85.0**	**90.0**	**90.0**	**85.0**	**80.0**	**90.0**	**95.0**	65.0	30.0	55.0	77.7
Polar question	20	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**95.0**	10.0	85.0	55.0	85.0	84.5
Scrambling	20	**90.0**	**75.0**	**70.0**	**85.0**	**85.0**	50.0	**65.0**	**85.0**	60.0	20.0	35.0	65.5
Topicalization	19	**89.5**	**78.9**	**78.9**	**78.9**	**78.9**	**78.9**	**68.4**	**84.2**	**73.7**	36.8	36.8	71.3
Wh-movement	35	91.4	**100.0**	91.4	**94.3**	**97.1**	91.4	**97.1**	8.6	**94.3**	57.1	71.4	81.3
MWE	80	**80.0**	**75.0**	**71.3**	**77.5**	**77.5**	**71.3**	**70.0**	**78.8**	**73.8**	45.0	37.5	68.9
Collocation	20	**100.0**	**90.0**	80.0	**95.0**	**95.0**	75.0	80.0	**95.0**	**90.0**	35.0	20.0	77.7
Idiom	20	**25.0**	**20.0**	**15.0**	**20.0**	**20.0**	**15.0**	**5.0**	**20.0**	**20.0**	0.0	**5.0**	15.0
Prepositional MWE	20	**95.0**	**95.0**	**95.0**	**95.0**	**95.0**	**95.0**	**95.0**	**100.0**	**90.0**	70.0	60.0	89.5
Verbal MWE	20	**100.0**	**95.0**	**95.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**95.0**	75.0	65.0	93.2
Named entity & terminology	89	**92.1**	**84.3**	**87.6**	82.0	82.0	**88.8**	**87.6**	**85.4**	68.5	70.8	73.0	82.0
Date	20	**100.0**	**100.0**	**100.0**	85.0	**100.0**	**100.0**	**100.0**	**95.0**	60.0	75.0	**100.0**	92.3
Domainspecific term	20	75.0	60.0	60.0	65.0	60.0	70.0	65.0	70.0	50.0	50.0	40.0	60.5
Location	20	95.0	95.0	95.0	95.0	90.0	95.0	90.0	95.0	90.0	85.0	90.0	92.3
Measuring unit	20	**100.0**	**90.0**	**100.0**	**90.0**	**90.0**	**100.0**	**100.0**	**90.0**	75.0	85.0	80.0	90.9
Proper name	9	**88.9**	**66.7**	**77.8**	**66.7**	**55.6**	**66.7**	**77.8**	**66.7**	**66.7**	**44.4**	33.3	64.6
Negation	20	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**95.0**	**100.0**	**100.0**	**95.0**	80.0	**100.0**	97.3
Non-verbal agreement	61	**91.8**	**88.5**	**88.5**	**86.9**	**90.2**	**83.6**	**82.0**	**88.5**	**85.2**	54.1	57.4	81.5
Coreference	20	80.0	70.0	75.0	80.0	75.0	70.0	70.0	75.0	70.0	70.0	60.0	72.3
External possessor	21	**100.0**	**95.2**	**90.5**	85.7	**95.2**	81.0	81.0	**90.5**	**90.5**	14.3	42.9	78.8
Internal possessor	20	**95.0**	**100.0**	**100.0**	**95.0**	**100.0**	**100.0**	**95.0**	**100.0**	**95.0**	80.0	70.0	93.6

phenomenon	items	Tohoku	VolcTrans	UEdin	Onl-B	Onl-G	Onl-A	PROMT	OPPO	Onl-Z	ZLabs	WMTBi	avg
Punctuation	60	**96.7**	**98.3**	**98.3**	71.7	61.7	**100.0**	**98.3**	70.0	28.3	68.3	55.0	77.0
Comma	20	**100.0**	**100.0**	**100.0**	**90.0**	**100.0**	**100.0**	**100.0**	**100.0**	85.0	85.0	85.0	95.0
Quotation marks	40	**95.0**	**97.5**	**97.5**	62.5	42.5	**100.0**	**97.5**	55.0	0.0	60.0	40.0	68.0
Subordination	180	**90.6**	**88.3**	**91.1**	**91.1**	**92.2**	**88.9**	**90.0**	**90.6**	**87.8**	65.0	62.2	85.3
Adverbial clause	20	90.0	90.0	90.0	90.0	95.0	90.0	90.0	85.0	90.0	85.0	80.0	88.6
Cleft sentence	20	**95.0**	**90.0**	**90.0**	**95.0**	**95.0**	**90.0**	**95.0**	**95.0**	**80.0**	55.0	**75.0**	86.8
Free relative clause	20	**90.0**	**95.0**	**95.0**	**95.0**	**90.0**	**90.0**	**85.0**	**90.0**	**85.0**	65.0	60.0	85.5
Indirect speech	20	**95.0**	85.0	**90.0**	85.0	**100.0**	**90.0**	**90.0**	**95.0**	80.0	50.0	70.0	84.5
Infinitive clause	20	**100.0**	**100.0**	**100.0**	**95.0**	**100.0**	**100.0**	**100.0**	**100.0**	**95.0**	80.0	65.0	94.1
Object clause	20	**95.0**	**95.0**	**95.0**	**100.0**	**100.0**	**95.0**	**95.0**	**100.0**	**95.0**	75.0	70.0	92.3
Pseudo-cleft sentence	20	**70.0**	**70.0**	**70.0**	**75.0**	**80.0**	**75.0**	**70.0**	**75.0**	**75.0**	40.0	35.0	66.8
Relative clause	20	**85.0**	**80.0**	**95.0**	**90.0**	**85.0**	**75.0**	**90.0**	**80.0**	**90.0**	**80.0**	65.0	83.2
Subject clause	20	**95.0**	**90.0**	**95.0**	**95.0**	85.0	**95.0**	**95.0**	**95.0**	**100.0**	55.0	40.0	85.5
Verb tense/aspect/mood	4447	**84.6**	**85.3**	80.3	75.9	79.6	77.5	75.1	79.3	73.6	50.5	52.1	74.0
Conditional	19	**100.0**	**100.0**	**100.0**	**100.0**	**89.5**	**89.5**	**94.7**	**94.7**	84.2	73.7	84.2	91.9
Ditransitive - future I	36	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	91.7	75.0	66.7	93.9
Ditransitive - future I subjunctive II	36	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	75.0	55.6	93.7
Ditransitive - future II	36	**100.0**	**100.0**	**100.0**	**100.0**	86.1	**100.0**	**97.2**	**100.0**	**100.0**	69.4	91.7	94.9
Ditransitive - future II subjunctive II	36	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**97.2**	**100.0**	**100.0**	**100.0**	69.4	**94.4**	96.5
Ditransitive - perfect	36	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**97.2**	83.3	86.1	97.0
Ditransitive - pluperfect	36	**94.4**	**94.4**	66.7	86.1	80.6	**100.0**	22.2	63.9	33.3	44.4	50.0	66.9
Ditransitive - pluperfect subjunctive II	36	**100.0**	91.7	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**97.2**	33.3	63.9	89.6
Ditransitive - present	36	**97.2**	**97.2**	**100.0**	**100.0**	**100.0**	**97.2**	72.2	**97.2**	83.3	75.0	63.9	89.4
Ditransitive - preterite	36	**88.9**	**86.1**	77.8	**97.2**	77.8	77.8	75.0	**94.4**	83.3	50.0	50.0	78.0
Ditransitive - preterite subjunctive II	36	**75.0**	**75.0**	**69.4**	**72.2**	**63.9**	**66.7**	**66.7**	**72.2**	**77.8**	36.1	44.4	65.4
Imperative	20	**85.0**	**80.0**	**80.0**	**95.0**	**95.0**	**75.0**	**80.0**	**90.0**	**85.0**	30.0	35.0	75.5
Intransitive - future I	36	**97.2**	**97.2**	**97.2**	**97.2**	**100.0**	**97.2**	**100.0**	**97.2**	**97.2**	86.1	66.7	93.9
Intransitive - future I subjunctive II	36	**100.0**	**100.0**	**100.0**	86.1	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	80.6	69.4	94.2
Intransitive - future II	42	88.1	90.5	**95.2**	**95.2**	88.1	92.9	**100.0**	**97.6**	92.9	61.9	35.7	85.3
Intransitive - future II subjunctive II	36	**100.0**	**100.0**	**100.0**	**100.0**	61.1	**100.0**	**100.0**	**100.0**	**100.0**	44.4	36.1	85.6
Intransitive - perfect	84	**100.0**	**100.0**	**100.0**	**100.0**	**98.8**	**100.0**	**100.0**	**100.0**	**97.6**	63.1	54.8	92.2
Intransitive - pluperfect	36	**83.3**	**83.3**	**66.7**	33.3	**83.3**	**75.0**	50.0	**77.8**	61.1	38.9	19.4	61.1
Intransitive - pluperfect subjunctive II	36	**100.0**	**100.0**	**97.2**	**94.4**	**100.0**	**100.0**	**94.4**	**97.2**	**100.0**	25.0	19.4	84.3
Intransitive - present	36	**100.0**	**100.0**	**100.0**	**97.2**	**100.0**	**100.0**	**100.0**	**100.0**	**97.2**	69.4	52.8	92.4
Intransitive - preterite	66	**93.9**	**97.0**	86.4	**92.4**	**92.4**	**95.5**	84.8	**97.0**	**95.5**	47.0	19.7	82.0
Intransitive - preterite subjunctive II	36	63.9	**75.0**	63.9	**77.8**	**80.6**	61.1	**88.9**	**75.0**	66.7	22.2	11.1	62.4
Modal - future I	180	79.4	**90.0**	73.9	78.9	79.4	73.9	72.2	75.6	75.6	62.2	59.4	74.6
Modal - future I subjunctive II	180	77.2	**86.7**	76.1	76.7	74.4	72.8	65.0	66.1	70.6	57.2	64.4	71.6
Modal - perfect	180	**90.0**	**83.9**	**85.0**	73.9	70.6	65.0	82.8	78.3	60.0	1.7	63.3	68.6
Modal - pluperfect	180	**50.6**	**40.6**	37.2	22.8	32.8	15.6	28.3	16.7	2.2	0.0	8.9	23.2
Modal - pluperfect subjunctive II	180	**65.6**	**60.6**	**60.0**	43.9	**59.4**	**56.7**	**60.0**	54.4	**55.0**	37.2	32.8	53.2
Modal - present	180	**97.8**	**96.1**	93.3	80.0	**94.4**	91.7	69.4	**95.0**	92.2	81.7	69.4	87.4
Modal - preterite	180	**98.3**	**99.4**	94.4	96.7	**98.9**	96.7	93.9	**98.9**	**97.8**	80.6	72.8	93.5

phenomenon	items	Tohoku	VolcTrans	UEdin	Onl-B	Onl-G	Onl-A	PROMT	OPPO	Onl-Z	ZLabs	WMTBi	avg
Modal - preterite subjunctive II	180	**73.9**	**78.9**	**75.0**	60.0	**75.6**	**78.9**	**77.8**	**77.2**	61.7	57.8	52.2	69.9
Modal negated - future I	180	80.0	**92.8**	75.6	78.3	79.4	76.1	75.6	79.4	77.2	66.7	67.2	77.1
Modal negated - future I subjunctive II	180	**78.3**	**85.6**	76.1	75.6	76.1	70.6	69.4	77.2	69.4	57.8	63.3	72.7
Modal negated - perfect	171	**95.9**	**93.6**	**94.2**	70.2	73.7	70.2	88.3	80.7	70.2	9.4	67.8	74.0
Modal negated - pluperfect	169	**36.1**	**22.5**	18.3	**32.0**	**32.5**	14.8	12.4	4.1	0.0	**0.6**	7.1	16.4
Modal negated - pluperfect subjunctive II	179	**71.5**	**70.9**	**73.7**	40.8	52.5	57.0	53.6	63.1	54.2	31.8	34.6	54.9
Modal negated - present	169	**100.0**	**100.0**	**99.4**	84.6	92.3	**98.8**	76.3	**99.4**	84.0	81.7	82.8	90.9
Modal negated - preterite	179	**99.4**	**100.0**	96.6	91.1	**98.9**	**99.4**	92.7	**100.0**	88.3	74.3	76.0	92.4
Modal negated - preterite subjunctive II	174	**74.7**	**81.0**	70.1	62.6	**79.9**	**81.6**	**75.3**	70.1	**75.9**	54.6	50.0	70.5
Progressive	19	**84.2**	**73.7**	**63.2**	**78.9**	**73.7**	**84.2**	**78.9**	**84.2**	**63.2**	15.8	36.8	67.0
Reflexive - future I	36	**100.0**	**100.0**	91.7	88.9	88.9	**94.4**	80.6	**94.4**	**97.2**	69.4	33.3	85.4
Reflexive - future I subjunctive II	36	83.3	**86.1**	83.3	**91.7**	80.6	83.3	69.4	**88.9**	**97.2**	55.6	30.6	77.3
Reflexive - future II	36	**94.4**	**100.0**	88.9	91.7	91.7	86.1	77.8	**94.4**	69.4	41.7	27.8	78.5
Reflexive - future II subjunctive II	36	**83.3**	**88.9**	**83.3**	**91.7**	61.1	**83.3**	75.0	**86.1**	**86.1**	30.6	25.0	72.2
Reflexive - perfect	36	**100.0**	**100.0**	**94.4**	**94.4**	**94.4**	91.7	86.1	91.7	91.7	41.7	33.3	83.6
Reflexive - pluperfect	36	91.7	**100.0**	83.3	86.1	91.7	80.6	80.6	**97.2**	91.7	30.6	19.4	77.5
Reflexive - pluperfect subjunctive II	36	**86.1**	**83.3**	**80.6**	**91.7**	**88.9**	**80.6**	75.0	**80.6**	**88.9**	36.1	22.2	74.0
Reflexive - present	36	**97.2**	**100.0**	91.7	91.7	91.7	86.1	69.4	**94.4**	88.9	33.3	13.9	78.0
Reflexive - preterite	36	**94.4**	**86.1**	69.4	**91.7**	**86.1**	69.4	75.0	**88.9**	75.0	13.9	16.7	69.7
Reflexive - preterite subjunctive II	36	**83.3**	**75.0**	61.1	**77.8**	**80.6**	61.1	**69.4**	**75.0**	**66.7**	22.2	19.4	62.9
Transitive - future I	42	**97.6**	**97.6**	**97.6**	**97.6**	**97.6**	**97.6**	**97.6**	**97.6**	**95.2**	**90.5**	81.0	95.2
Transitive - future I subjunctive II	36	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**97.2**	77.8	97.7
Transitive - future II	36	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	91.7	63.9	96.0
Transitive - future II subjunctive II	36	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**97.2**	63.9	96.5
Transitive - perfect	42	**100.0**	**100.0**	**100.0**	**100.0**	**97.6**	**100.0**	**100.0**	**100.0**	**95.2**	85.7	66.7	95.0
Transitive - pluperfect	36	**100.0**	**100.0**	77.8	80.6	**100.0**	**100.0**	75.0	**94.4**	80.6	66.7	36.1	82.8
Transitive - pluperfect subjunctive II	36	**100.0**	91.7	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	30.6	44.4	87.9
Transitive - present	48	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**95.8**	**100.0**	**97.9**	75.0	75.0	94.9
Transitive - preterite	36	**97.2**	**97.2**	88.9	91.7	**97.2**	88.9	91.7	**100.0**	**97.2**	66.7	55.6	88.4
Transitive - preterite subjunctive II	36	**66.7**	**75.0**	**63.9**	**69.4**	61.1	58.3	**80.6**	**83.3**	**69.4**	25.0	50.0	63.9
Verb valency	87	**79.3**	**81.6**	**77.0**	**81.6**	**77.0**	**77.0**	**71.3**	**80.5**	64.4	44.8	51.7	71.5
Case government	28	**92.9**	**96.4**	**85.7**	**92.9**	**82.1**	**89.3**	78.6	**92.9**	**82.1**	42.9	57.1	81.2
Mediopassive voice	20	**85.0**	**90.0**	**85.0**	**95.0**	**90.0**	**80.0**	70.0	**90.0**	55.0	50.0	50.0	76.4
Passive voice	19	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**89.5**	78.9	73.7	94.7
Resultative predicates	20	35.0	35.0	35.0	35.0	35.0	35.0	35.0	35.0	25.0	10.0	25.0	30.9
micro-average	5514	**85.3**	**85.4**	81.2	77.7	80.6	78.7	76.5	79.1	73.6	51.3	52.4	74.7
phenomenon macro-average	5514	**89.1**	88.0	85.2	85.1	85.5	83.9	82.1	83.7	78.5	53.7	51.8	78.8
category macro-average	5514	**88.1**	86.8	85.3	84.6	84.3	83.6	82.7	80.0	74.1	54.9	54.4	78.1
BLEU		43.8	43.5	42.3	41.9	41.4	40.4	39.6	43.2	35.4	31.5	32.1	35.4

Table 7: Accuracies (%) of successful translations for 11 systems regarding all phenomena, organized in categories. Boldface indicates the best scoring system in each row, including all systems which are not significantly inferior than the best scoring system. Grey rows average the accuracies of the phenomena per category.

Gender Coreference and Bias Evaluation at WMT 2020

Tom Kocmi *
Microsoft
tomkocmi@microsoft.com

Tomasz Limisiewicz
Charles University in Prague
Faculty of Mathematics and Physics
limisiewicz@ufal.mff.cuni.cz

Gabriel Stanovsky
The Hebrew University of Jerusalem
gabis@cse.huji.ac.il

Abstract

Gender bias in machine translation can manifest when choosing gender inflections based on spurious gender correlations. For example, always translating doctors as men and nurses as women. This can be particularly harmful as models become more popular and deployed within commercial systems. Our work presents the largest evidence for the phenomenon in more than 19 systems submitted to the WMT over four diverse target languages: Czech, German, Polish, and Russian. To achieve this, we use WinoMT, a recent automatic test suite which examines gender coreference and bias when translating from English to languages with grammatical gender. We extend WinoMT to handle two new languages tested in WMT: Polish and Czech. We find that all systems consistently use spurious correlations in the data rather than meaningful contextual information.

1 Introduction

Bias in machine learning occurs when systems pick up correlations which are useful for specific training *datasets*, but are not indicative for the *task* that the dataset represents.

In the context of machine translation (MT), gender bias can occur when translating from languages without grammatical noun genders, such as English or Turkish, to a language with gender inflections, such as Spanish, Polish, or Czech. In such cases, the translation model needs to assign gender inflection in the target language based on contextual cues in the source text. For example, when translating the English sentence "The *doctor* asked the nurse to help *her* in the operation" to Spanish, the model needs to produce the female inflected "doctora" based on the feminine English pronoun "her".

Recently, Stanovsky et al. (2019) created a challenge set and an automatic evaluation metric, dubbed WinoMT, to examine whether popular MT models are capable of correctly capturing and translating such information from English into a diverse set of 8 target languages with grammatical gender. They found that all six tested systems, composed of four commercial and two academic models, consistently relied on gender role assignments in the data regardless of context. In our example above, models would prefer to translate the doctor using masculine inflections, despite the context suggesting otherwise.

In this work, we apply the WinoMT test suite on the submissions to the News shared task of WMT 2020. In addition to testing the phenomenon on a large number of models, we extend the WinoMT to the Polish and Czech languages, tackling unique language-specific challenges. We thoroughly analyze the extent of the phenomena for the tested languages and systems, as well as its correlation with the widely-used BLEU evaluation metric (Papineni et al., 2002), finding that systems with worse performance (in BLEU) make more errors for female professions than errors for male professions. On the other hand, better-performing systems (in BLEU) make more errors related to anti-stereotypical professions (e.g. female doctors, or male nurses).

Similarly to the conclusions of Stanovsky et al. (2019), we find that all systems consistently perform better when the source texts exhibit stereotypical gender role assignments (e.g., male doctors, female nurses) versus non-stereotypical assignments (e.g., female doctors, male nurses), indicating that these models rely on spurious correlations in their training data, rather than on more meaningful textual context. We hope that this evaluation will be used as a standard evaluation metric for MT as a means to track the improvement of this socially important aspect of translation.

* Part of work performed while at Charles University.

Proceedings of the 5th Conference on Machine Translation (WMT), pages 357–364
Online, November 19–20, 2020.

2 Background: WinoMT

WinoMT was created as a concatenation of two coreference test suites: WinoGender (Zhao et al., 2018a) and WinoBias (Rudinger et al., 2018). Each instance in these datasets is a single English sentence, presenting two entities, identified by their profession (e.g., "teacher", "janitor", or "hairdresser") and a single pronoun referring to one of them based on the context of the sentence. For example, in the sentence "The physician hired the secretary because *he* was overwhelmed with clients", the marked pronoun refers to the physician. In contrast, in "The physician hired the secretary because *he* had good credentials" the pronoun likely refers to the secretary. Both datasets are created with an equal amount of stereotypical gender role assignments (e.g., the first example) and non-stereotypical assignments (e.g., the second example). Both works found that coreference systems performed much better on the stereotypical role assignments than they did on the non-stereotypical ones, concluding that systems relied on training correlations between pronoun gender and professions rather than the syntactic and semantic information in the input sentence.

Stanovsky et al. (2019) use these two corpora to test gender bias in machine translation in the following manner:

1. An MT model is used to translate these corpora into a target language with grammatical gender.

2. A language-specific, target-side morphological analysis identifies the gender of the translated entity (e.g., the physician in the first example above).

3. The gold and predicted genders are compared between the English and target sentence.

Following this procedure, they tested four commercial systems and two state-of-the-art academic models on eight diverse target languages: Spanish, French, Italian, Hebrew, Arabic, Ukrainian, Russian, and German. In all of their experiments, they found that similarly to coreference models, MT systems are prone to make gender-biased predictions.

In Section 3, we describe our extension of WinoMT to two additional languages tested in WMT 2020: Czech and Polish, and in Section 4 we use the extended test suite to evaluate WMT submission on a total of four target languages: Czech, German, Polish, and Russian.

3 New Target Languages: Polish and Czech

In this section, we describe our extension of WinoMT for Czech and Polish. The methods and analyses for both target languages are done by the first two authors, who are native speakers in the respective language.

For both languages, we followed the approach of WinoMT, where translated sentences are first aligned by fast_align (Dyer et al., 2013), followed by automatic morphology analysis.

Besides, we notice that the automatic alignment and existing tools sometimes fail leading to "unknown" gender decision. For both Czech and Polish, it could not recognise on average 10–15% test examples.

Fortunately, both languages have rich morphology where gender can be often identified from the word form. Therefore, we have created a list of the most often translations of each profession in all cases. Example of such a list is in Table 1. We use this list to the first check if the gender can be recognised solely based on the word form. In case that the word is not in our predefined list or if both the male and female version are possible. We revert to language-specific automatic analysis, as described below.

This step significantly reduced the number of unrecognised genders. In Section 3.4, we discuss the number of unrecognised genders of the profession.

3.1 Czech analysis

For translated professions in Czech that were not resolved by the predefined list, we use the automatic morphology tagger MorphoDiTa (Straková et al., 2014). This tool uses a morphological dictionary and estimates regular patterns based on common form endings, by which it clusters morphological "templates" without linguistic knowledge of Czech.

When analysing Czech, we ignore all examples that test neutral form, as Czech does not use neutral case as a grammatical structure allowing both genders.[1] Additionally, we ignore a few idiosyncratic edge cases: The word "advisee" cannot be directly

[1] In a few cases, the neutral form can be created by inaccurate translation, when replacing a profession with a place, where the professional works. For example, "hairdresser" can be replaced by "hair saloon" which is neutral in Czech.

Profession	Gender	Possible forms
Chef	Male	šéfce, šéfka, šéfko, šéfkou, šéfku, šéfky, náčelnice, náčelnici, náčelnicí, ...
Chef	Female	šéf, šéfa, šéfe, šéfem, šéfovi, šéfu, náčelník, náčelníka, náčelníkem, ...
Cashier	Male	pokladní, pokladního, pokladním, pokladnímu
Cashier	Female	pokladní

Table 1: Example of a list of possible forms for a given profession and gender (some forms are missing). Some professions have several possible translations; in this example "chef" has two possible translations. In Czech, most of the forms are distinct between male and female form. However, it is not always the case as can be seen for example for "cashier", where both male and female can have the form "pokladní". In those cases, we need to rely on automatic annotation based on the context of the whole sentence.

translated into Czech, while "guest" and "mover" do not have a female counterpart.

Altogether, we exclude 470 examples from WinoMT, reducing its size for Czech analysis to 3418 examples.

Lastly, certain translated professions have the same form for both male and female, for example, the word "vedoucí" ("supervisor", either male or female). In such cases, our analysis cannot correctly assign correct gender. Therefore we mark these example as a correct with the use of the gold data.

3.2 Polish analysis

For translated professions in Polish that are not found in the prepared list of possible word forms, we conduct an automatic morphological tagging to find their grammatical gender. For that purpose, we use a recently released spaCy model (Honnibal and Montani, 2017) with tagger for Polish (Tuora and Kobyliński, 2019), which relies on dictionary-based morphology analysis performed by Morfeusz (Woliński, 2014).

Similarly to Czech, in Polish, there are no names of professions with a neutral gender. Therefore for Polish analysis, we also ignore test cases for neural form. Additionally, we do not evaluate gender for the professions that do not have a polish translation, i.e. "advisee" and "mover". This reduces the Polish testset to 3136 examples.

In Polish, it is possible to indicate gender for almost all profession names. In most cases, it can be formed by changing the suffix of the word. Nevertheless, for specific occupations, female counterparts created by derivation are rarely used and do not appear in major language dictionaries. For such professions, a feminine variant is obtained by adding a word indicating gender in front – usually "pani(ą)" ("mrs.") before the masculine form of the occupation name. In our evaluation, we accept both variants.

We have identified 16 professions without feminine derivations in the on-line version of the Grammatical Dictionary of Polish (Woliński and Kieraś, 2016). These words are: "appraiser", "driver", "electrician", "engineer", "firefighter", "investigator", "mechanic", "pathologist", "plumber", "scientist", "sheriff", "surgeon", "taxpayer", "veterinarian", "witness", and "guest". We decided to keep these test cases because we observed a few interesting examples of correctly translating gender for them (as discussed in section 4.3) and see a potential for further improvement.

3.3 Human Annotation

We conducted a human evaluation of gender bias for the two new languages. We sampled 300 instances from the output of all systems; each sample was annotated by two Czech and two Polish native-speakers, with a third annotator resolving differences. Following the human evaluation protocol of Stanovsky et al. (2019), annotators were shown an entity in English and the translated sentence. They were asked to provide the gender of the entity in the target language.

We then compared human annotations with the output of our morphological analysers in both languages. The inter-annotator agreement was high: 96.3% for Czech and 98.6% for Polish. Finally, the performance of both system was good enough to support further analyses — the Czech analyser achieved 96.3% accuracy, while the Polish analyser achieved 98.8%. Both of these numbers surpass the average performance reported in (Stanovsky et al., 2019) of 87%.

Furthermore, our whitelisting approach for Czech can resolve almost all cases. From our WMT20 testsuite evaluation, it resolved all but

284 sentences out of 46,656 evaluated by all systems. We conducted a human annotation over these 284 sentences and found out that our automatic approach can correctly resolve 64.3 % examples. Likewise for Polish, whitelist approach failed in only 1,533 out of 47,267 examples, where it needed to rely on the morphology evaluation. We selected 300 random sentences from this subset for human evaluation. The agreement between our morphology algorithm and annotators was 78.7%. We have to stress that those are the most challenging examples and most often incorrect translations.

3.4 Unrecognized Gender

When neither whitelisting nor contextual morphological analysis recognise gender properly, our automatic assigns an "unknown" gender. This happens mostly for erroneous translations, when the translation does not contain the profession at all (for example when "hairdresser" is translated as a "hair salon"), or when an error is made by the alignment or morphology annotation.

We consider two approaches for handling such unrecognised genders. We could ignore examples with "unknown" gender, or we can count them as errors. The former approach would change the ratio between male and female professions differently for each system; in other words, each system would have a testset of different size based on its performance. The latter approach, on the other hand, will punish systems for an error is in the analysis. We follow WinoMT, where the latter approach is selected.

To estimate the implication of this choice, in Table 2 we present the average percentage of "unknown" genders when a gold label is male or female. The percentage is averaged across all systems. We can notice that number of unknowns is a minimal form all languages except Russian, and the difference between unrecognised male and female professions is small. Therefore, it should not skew the results of the analysis.

We observe that the errors are usually due to the translation issue. For example, in Czech, the system with the most unrecognised genders is also the worst-performing in terms of BLEU. This system ("zlabs-nlp") has 248 unrecognised professions out of the whole testset, while the average for all systems is 90, and it has a performance of 20.3 BLEU score, while the second-worst system has a performance of 25.3 BLEU.

Target Language	Unknown Male	Unknown Female
Czech	1.33%	1.30%
German	1.61%	1.38%
Polish	1.37%	1.84%
Russian	12.89%	13.53%

Table 2: The percentage of "unknowns" for gold male or female labels, averaged across all submissions for a target language.

The Russian analysis cannot recognise more than 12% of professions. We believe that this could be improved in the future with the whitelisting approach as was done for Czech and Polish.

4 Evaluation

We continue with the analysis as described by Stanovsky et al. (2019). For each system, we compute three metrics that represent their ability to resolve gender coreference; or how often the systems resolve the gender-based on stereotypical genders of professions. All results are in Table 3.

4.1 Results

Overall accuracy. First, the overall system Accuracy (abbreviated as "Acc") is calculated as a percentage of instances in which the translation preserved the gender of the profession from the original English sentence. We find that most systems perform better than random guessing. One exception is the Russian language, where all systems perform worse than random guessing. This could be related to a problematic analysis as mentioned in Section 3.4, where all systems are penalised for "unknown" genders, which results in lowering their accuracy. Overall, the system with the best accuracy is CUNI-DocTransformer on the Czech language. This system has been trained on a document-level instead of separate sentences, which may have helped it learn to resolve coreference better than sentence-level systems. Among systems that participated in all four languages, *OPPO* performs the best, also outperforming commercial systems (anonymised as "online-X").

Gender-based performance analysis. Second, we compute the difference Δ_G in performance (F1 score) between male and female translated professions. Δ_G=0 means that the system makes an equal number of errors on both male and female professions. This should be the correct case in ideal conditions as there is an equal number of male and

Translation System	Czech			German			Polish			Russian		
	Acc	Δ_G	Δ_S	Acc	Δ_G	Δ_S	Acc	Δ_G	Δ_S	Acc	Δ_G	Δ_S
OPPO	78.7	4.7	30.0	**75.9**	-1.9	16.9	68.2	14.5	28.4	43.2	28.1	12.2
zlabs-nlp	49.9	38.3	16.3	71.9	1.1	8.5	46.1	50.3	4.3	36.3	37.8	6.7
eTranslation	70.9	11.0	34.5	71.3	2.9	18.0	68.8	11.8	29.0	-	-	-
SRPOL	81.2	3.4	24.3	-	-	-	**71.2**	12.0	27.6	-	-	-
CUNI-Transformer	78.0	5.6	31.8	-	-	-	69.8	14.1	30.6	-	-	-
CUNI-DocTransformer	**83.6**	2.2	22.7	-	-	-	-	-	-	-	-	-
CUNI-T2T-2018	77.6	5.5	28.1	-	-	-	-	-	-	-	-	-
UEDIN-CUNI	72.5	9.4	28.9	-	-	-	-	-	-	-	-	-
AFRL	-	-	-	69.7	5.8	14.7	-	-	-	-	-	-
Tohoku-AIP-NTT	-	-	-	70.4	1.3	23.2	-	-	-	-	-	-
UEDIN	-	-	-	66.6	9.0	18.7	-	-	-	-	-	-
WMTBiomedBaseline	-	-	-	49.5	34.5	5.9	-	-	-	-	-	-
Huoshan_Translate	-	-	-	63.8	8.0	24.5	65.7	18.5	30.7	-	-	-
PROMT_NMT	-	-	-	65.7	7.2	17.7	-	-	-	44.3	23.8	14.0
NICT_Kyoto	-	-	-	-	-	-	64.2	19.6	32.2	-	-	-
SJTU-NICT	-	-	-	-	-	-	68.2	15.6	26.1	-	-	-
Tilde (1425)	-	-	-	-	-	-	63.3	19.1	32.3	-	-	-
Tilde (1430)	-	-	-	-	-	-	64.8	17.7	23.2	-	-	-
ariel197197	-	-	-	-	-	-	-	-	-	34.1	29.6	15.5
online-a	63.3	21.7	21.7	74.5	0.1	12.5	53.7	37.8	21.9	39.1	35.9	10.2
online-b	56.9	29.7	19.2	68.3	2.9	19.4	57.7	31.9	21.3	37.8	36.9	10.4
online-g	62.0	22.5	25.9	62.2	12.0	16.0	67.3	17.5	27.7	**47.7**	16.2	17.5
online-z	72.2	8.2	30.9	73.6	0.6	12.4	65.9	16.0	35.1	44.4	25.6	12.5

Table 3: The evaluation on WinoMT testset for translation systems submitted to WMT 2020. *Acc* indicates overall gender accuracy (% of instances the translation had the correct gender), Δ_G denotes the difference in performance (F1 score) between masculine and feminine scores, and Δ_S is the difference in accuracies between pro-stereotypical and anti-stereotypical gender role assignments.

female examples in WinoMT. Positive Δ_G indicates that the system makes fewer errors for male professions and more errors with female professions. Almost all systems perform significantly better on male professions. This could be a result of training data that contains more male examples than female ones. However, we observe that many systems have Δ_G close to zero. An interesting situation is in Czech analysis, where there is a broad range of Δ_G values.

Stereotypical vs non-stereotypical examples Third, we measure the difference Δ_S in performance (F1 score) between stereotypical and non-stereotypical gender role assignments. The stereotypicality of the profession was determined based statistics provided by the US Department of Labor (see Zhao et al. (2018b)). Δ_S in Table 3 shows that all systems have a significantly better performance when presented with pro-stereotypical assignments (e.g., a female nurse), while their performance deteriorates when translating anti-stereotypical roles (e.g., a male receptionist). These analyses indicate that all MT systems are gender-biased, prone to translate gender inflexions based on training set correlations rather than contextual cues in specific input instances.

4.2 Gender Bias vs BLEU

Another interesting comparison is between the overall translation performance of a system and its observed gender bias. Unfortunately, the official WMT human annotation was not available to us at the time of writing. Instead, we evaluate the performance of all systems with the automatic BLEU metric (Papineni et al., 2002). The evaluation is done with official WMT20 testset (Barrault et al., 2020) and the SacreBLEU implementation (Post, 2018).[2]

In Fig. 1 we present pairwise relationships and correlations between our metrics (Accuracy, Δ_G, Δ_S) and BLEU. We observe that correlation between gender accuracy and BLEU is moderately strong (Pearson's ρ 0.66). There is a significant negative association between Δ_G and both gender accuracy and BLEU, meaning that systems scoring high in those metrics perform similarly well on male and female examples. We observe a low positive correlation between BLEU and Δ_S and a moderate positive correlation between gender

[2]SacreBLEU signature is: BLEU+case.mixed+numrefs.1 +smooth.exp+tok.13a+version.1.4.14

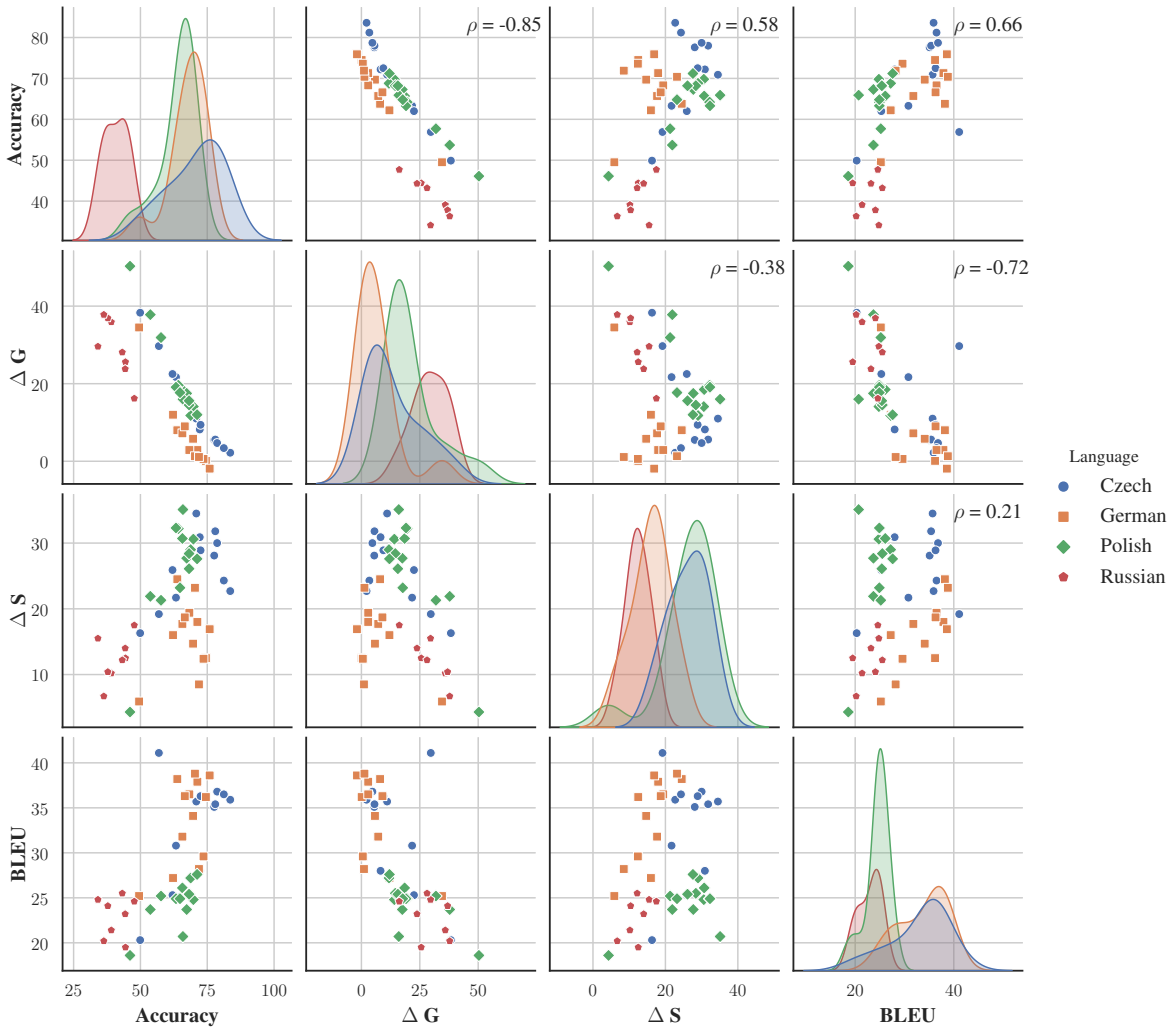


Figure 1: Relationships between gender accuracy, ΔG, ΔS and BLEU. Each point represents evaluation of one system's output. ρ in the upper right is Pearson's correlation coefficient between row and column variables.

accuracy and Δ_S. It implies that the system well-performing in the automatic evaluation may overfit to examples with stereotypical gender.

4.3 Qualitative Analysis

In our analysis of new target languages: Czech and Polish, we have observed several linguistic phenomena that affect gender bias of translations. We illustrate them with exemplary translations of WinoMT test sentences in Table 4.

In Polish, feminine forms of some words are not commonly used, e.g., "mechanic" and "veterinarian". We provide a complete list of occupations is in Section 3.2. According to our expectations, such words are especially problematic for evaluated systems. The female gender was correctly assigned in only 1.5% of the translations. Interestingly, in some cases, a word indicating female gender is added, even though it is not used in a source sentence (see in Table 4), such a translation is marked as correct.

5 Conclusions

We have extended analysis by Stanovsky et al. (2019) with Czech and Polish languages.

We showed that current systems, both commercial and academic, perform worse in gender coreference when profession in question is female. This is amplified if the system in question has lower translation quality.

Moreover, systems rely for translations on stereotypical genders of professions instead of correct gender coreference resolution. We showed that with increasing translation quality, models make fewer errors in general, but rely more often on the stereotypical genders.

Source	[Target lang.] Translation	Phenomenon
The librarian waved at the **mechanic** and walked away from **her**.	[PL] Bibliotekarka pomachała do **mechanika** i od **niej** odeszła.	Female form for "mechanic" in Polish is rarely used, hence it did not occur in output of the evaluated translation systems.
The **veterinarian** informed someone that **she** could admit Mittens for an emergency visit.	[PL] **Pani weterynarz** poinformowała właściciela, że może przyjąć Mitens na nagłą wizytę.	Word "weternenarz" has a masculine gender, but translation correctly captures gender by adding word "pani" ("mrs.").
The **chief** thanked the **nurse** and handed **him** a present.	[CS] **Sestřička** poděkovala **primáři** a dala **jí** dárek.	There is a missmatch in translation, as female form "jí" refers to male "primáři". Moreover, it swapped actors in the translation.
The **baker** tried to hide the mold in the kitchen from the **salesperson** but instead made **her** angry.	[CS] **Prodavač** si všiml **pekařovy** nechutné kuchyně a řekl **mu**, aby přestal prodávat chleba.	In this translation, the profession is in adjective, which we are unable to detect by our analysis.

Table 4: Examples of interesting linguistic phenomena observed for translation of WinoMT sentences into new target languages. Words in **blue**, **red**, and **orange** indicate male, female and neutral entities, respectively.

This is mainly a problem with the MT training data, that usually contain more examples of stereotypical professions in contrast to anti-stereotypical ones.

Acknowledgments

This study was supported in parts by the grants 18-24210S of the Czech Science Foundation and 825303 (Bergamot) of the European Union. This work has been using language resources and tools stored and distributed by the LINDAT/CLARIN project of the Ministry of Education, Youth and Sports of the Czech Republic (LM2015071).

We also also would like to thank Joanna Wetesko and Maciej Biesek for discussions about the role of grammatical gender in Polish and their help with the evaluation.

References

Loïc Barrault, Magdalena Biesialska, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Tom Kocmi, Philipp Koehn, Nikola Ljubešić, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. 2020. Findings of the 2020 conference on machine translation (wmt20). In *Proceedings of the Fifth Conference on Machine Translation, Volume 2: Shared Task Papers*. Association for Computational Linguistics.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648, Atlanta, Georgia. Association for Computational Linguistics.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting bleu scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

Rachel Rudinger, Jason Naradowsky, Brian Leonard, and Benjamin Van Durme. 2018. Gender bias in coreference resolution. *arXiv preprint arXiv:1804.09301*.

Gabriel Stanovsky, Noah A. Smith, and Luke Zettlemoyer. 2019. Evaluating gender bias in machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1679–1684, Florence, Italy. Association for Computational Linguistics.

Jana Straková, Milan Straka, and Jan Hajič. 2014. Open-Source Tools for Morphology, Lemmatization, POS Tagging and Named Entity Recognition. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 13–18, Baltimore, Maryland. Association for Computational Linguistics.

Ryszard Tuora and Łukasz Kobyliński. 2019. Integrating Polish language tools and resources in Spacy. In *Proceedings of PP-RAI 2019 Conference*, pages 210–214, Wrocław. Department of Systems and Computer Networks, Faculty of Electronics, Wroclaw University of Science and Technology.

Marcin Woliński and Witold Kieraś. 2016. The online version of grammatical dictionary of polish. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 2589–2594, Portorož, Slovenia. European Language Resources Association (ELRA).

Marcin Woliński. 2014. Morfeusz reloaded. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014*, pages 1106–1111, Reykjavík, Iceland. European Language Resources Association (ELRA).

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2018a. Gender bias in coreference resolution: Evaluation and debiasing methods. *arXiv preprint arXiv:1804.06876*.

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2018b. Gender bias in coreference resolution: Evaluation and debiasing methods. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 15–20, New Orleans, Louisiana. Association for Computational Linguistics.

The MuCoW word sense disambiguation test suite at WMT 2020

Yves Scherrer **Alessandro Raganato** **Jörg Tiedemann**
University of Helsinki
{name.surname}@helsinki.fi

Abstract

This paper reports on our participation with the MuCoW test suite at the WMT 2020 news translation task. We introduced MuCoW at WMT 2019 to measure the ability of MT systems to perform word sense disambiguation (WSD), i.e., to translate an ambiguous word with its correct sense. MuCoW is created automatically using existing resources, and the evaluation process is also entirely automated. We evaluate all participating systems of the language pairs English → Czech, English ↔ German, and English → Russian and compare the results with those obtained at WMT 2019. While current NMT systems are fairly good at handling ambiguous source words, we could not identify any substantial progress – at least to the extent that it is measurable by the MuCoW method – in that area over the last year.

1 Introduction

At WMT 2019, we introduced the MuCoW (*multilingual contrastive word sense disambiguation*) test suite (Raganato et al., 2019) and evaluated the news task submissions of nine translation directions with it.[1] We observed that systems generally performed quite well on word sense disambiguation, but found a big gap between in-domain and out-of-domain disambiguation performance for some translation directions, in particular with constrained systems.

For WMT 2020, we reuse the same test suite for the same language pairs. This gives us the opportunity to measure the advancement of machine translation within a year. We expect the larger training data sets and the model improvements to have a small but positive impact on translation quality in general, and word sense disambiguation performance in particular.

[1]The MuCoW test suite is available at http://github.com/Helsinki-NLP/MuCoW.

2 The MuCoW test suite

MuCoW (Raganato et al., 2019) is a language-independent method for automatically building test suites to assess the capabilities of MT systems to disambiguate between ambiguous words in the source language. The version of MuCoW used for WMT 2019 involves the following steps:

1. Identify ambiguous source nouns and their translations, using word-aligned and tagged parallel corpora from the OPUS collection (Tiedemann, 2012).

2. Cluster the translations into senses. First, we query BabelNet (Navigli and Ponzetto, 2012), a wide-coverage multilingual encyclopedic dictionary, to assign senses (synsets) to words. Second, we refine the results with the SW2V sense embeddings (Mancini et al., 2017).

3. Select sentences with ambiguous words and assign them sets of correct and incorrect target translations.

We evaluated the systems participating in the WMT 2019 news translation task with MuCoW for the language pairs English → Czech, English ↔ German, English ↔ Finnish, English ↔ Russian, and English ↔ Lithuanian.

A substantial amount of MuCoW sentences and senses come from the OpenSubtitles2018 corpus, but most systems participating at WMT are tuned towards the news domain and therefore are not expected to handle lexical choices of colloquial speech reliably. Therefore, we distinguished between in-domain and out-of-domain synsets: a synset is considered out-of-domain if more than half of its example sentences come from movie subtitles.

Proceedings of the 5th Conference on Machine Translation (WMT), pages 365–370
Online, November 19–20, 2020.

Example containing **ambiguous word**	Correct translations	Incorrect translations
It occurred to me that my **watch** might be broken.	Armbanduhr, Uhr	*Wache*
I hope you didn't get distracted during your **watch**.	*Wache*	Armbanduhr, Uhr
In winter, the dry leaves fly around in the **air**.	Luft, Luftraum, Aura	Miene, Ausdruck
He remained silent for a moment, with a thoughtful but contented **air**.	Miene, Ausdruck	Luft, Luftraum, Aura
Harry had to back out of the competition because of a broken **arm**.	Arm	*Waffe*
So does the cop who left his side **arm** in a subway bathroom.	*Waffe*	Arm
Drain the pasta and return the pasta to the **pot**.	Blumentopf, Kochtopf, Topf, Nachttopf	*Marihuana, Gras*
Where did those idiots get all of this **pot** anyhow?	*Marihuana, Gras*	Blumentopf, Kochtopf, Topf, Nachttopf

Table 1: Examples of test suite instances of the English–German test suite. The ambiguous (English) source word is highlighted in bold, and correct and incorrect (German) translations – as inferred by the MuCoW procedure – are given. Senses classified as out-of-domain are shown in italics. Note that some example sentences may further restrict the set of correct translations.

Language pair	Source words	Target synsets	In-dom synsets	Out-dom synsets	Sentences
EN–CS	98	200	29	171	1843
EN–DE	176	362	220	142	3337
DE–EN	217	461	329	132	4268
EN–RU	97	199	40	163	1814

Table 2: Sizes of the MuCoW data sets compiled for WMT 2019 and 2020.

In Raganato et al. (2020), we report on an extended version of MuCoW that covers the following aspects:

- The selection of data sources is improved to reduce noise and domain effects.
- The sense inference process is streamlined and relies on lemmatization instead of word alignment, leading to better coverage especially for morphologically rich languages.
- In addition to test sets, the composition of training data is also defined to guarantee that competing translation models are evaluated on fair grounds.

Since it was not possible to restrict the training data of participating WMT systems, we decided to reuse the WMT 2019 version again for WMT 2020, with exactly the same sentences. This allows us to trace the year-over-year evolution of translation quality with respect to lexical disambiguation. Therefore, the MuCoW analysis is restricted to the language pairs and translation directions that were already part of the WMT news task in 2019, namely English → Czech, English ↔ German, and English → Russian.

MuCoW data sets are created specifically for each language pair and translation direction (for details, see Raganato et al., 2019). Each entry consists of a sentence in the source language, the ambiguous source word, a list of correct target words (the correct target synset), a list of incorrect target words (the incorrect target synset), and information about the domain of the synsets. The participants only see the source sentences, not the metadata. Table 1 shows a few example sentences taken from the English–German test suite. The main statistics of the test suites used for WMT 2020 are reported in Table 2.

3 Evaluation and Results

The source language sentences were sent to the WMT participants as part of the test set, and we received the translations in the target language for evaluation. We then checked if any of the correct or incorrect target words listed in the metadata file could be identified in the translation output.

Although the sentences were selected to contain the uninflected base forms both in the source and target languages, we could not assume that all translation systems would output base forms. Hence, if neither correct nor incorrect target words could be identified in the tokenized translations, we lemmatized them and searched the target words again in the lemmatized version.[2] Depending on the morphological properties of the target language, lemmatization substantially increased the coverage (see Table 3). Between 2019 and 2020, the average coverage has remained constant

[2]We used the Turku neural lemmatizer with pretrained models (Kanerva et al., 2019).

Language pair	Avg. coverage (tokenized)	Avg. coverage (tok. + lemmatized)
EN–CS	63.16%	75.82%
	61.77%	*74.87%*
EN–DE	69.43%	72.08%
	66.52%	*69.26%*
DE–EN	83.10%	84.41%
	83.06%	*84.51%*
EN–RU	65.13%	80.13%
	58.88%	*73.29%*

Table 3: Average coverage of target words among WMT 2019 (in gray italics) and WMT 2020 (in black) primary submissions.

for DE–EN, slightly increased for EN–CS and EN–DE, and substantially increased for EN–RU. We assume that these increases are mostly due to the different number and composition of the submissions.

We report precision, recall and F1-score for in-domain senses and out-of-domain senses separately. Precision and recall are computed as follows:[3]

$$\text{Precision} = \frac{\text{\# examples with correct target words}}{\text{\# examples with either correct or incorrect target words}}$$

$$\text{Recall} = \frac{\text{\# examples with correct target words}}{\text{\# total examples}}$$

The results are shown in Tables 4 to 7, with WMT 2019 and 2020 submissions side-by-side.

For all four examined translation directions, the best 2019 results were beaten in 2020. However, one of the best-performing systems in 2019, *Facebook_FAIR*, did not participate in 2020. The *Facebook_FAIR* system is characterized by high precision rates, whereas the winning 2020 systems (such as *Tohoku-AIP-NTT* or *Online-G*) benefit from higher recall. This shift suggests that the denominator of the precision computation comes closer to the one of the recall computation, or in other words that the translations themselves become more accurate. Further analysis will be required to substantiate this claim.

Interesting year-over-year comparisons can be observed for the *Online-G* system: it produces almost identical results in both years for English–German and English–Russian, but shows substantial improvements for the German–English direction.

[3]Examples that contained both correct and incorrect target words were counted as incorrect.

The overall result distributions show a slight upward trend in WSD performance for English–German and German–English, but less so for English–Czech and English–Russian. Since the participating systems differed over the years, it is of course difficult to draw any reliable conclusions.

For most language pairs, the in-domain and out-of-domain synsets produce similar rankings. Just like in 2019, English–Czech is an exception, where – contrarily to all expectations – an online system shows the best in-domain performance and a research system the best out-of-domain performance.

4 Conclusion

In this paper, we report our participation with the MuCoW test suite at the WMT 2020 news translation task. MuCoW is an automatically built WSD test suite for machine translation that relies on large parallel corpora, the multilingual lexical resource BabelNet and language-independent synset embeddings.

We find that state-of-the-art NMT systems are fairly good at handling ambiguous source words, but that no substantial progress – at least to the extent that it is measurable by the MuCoW method – has been made in that area over the last year. Among the top-performing systems, we observe a shift from high precision to high recall, hinting at general improvements in translation quality. It will therefore be particularly instructive to see how well the WSD test suite results correlate with human evaluation scores and with recently proposed evaluation metrics that are based on semantic representations of the translations (Gupta et al., 2015; Shimanaka et al., 2018).

Acknowledgments

This work is part of the FoTran project, funded by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 771113).

The authors gratefully acknowledge the support of the CSC – IT Center for Science, Finland, for computational resources.

English–Czech	In-domain synsets			Out-of-domain synsets			All synsets		
Submission	Prec.	Recall	F1	Prec.	Recall	F1	Prec.	Recall	F1
SRPOL	97.15	84.45	90.36	**80.38**	**73.78**	**76.94**	**83.22**	**75.67**	**79.27**
CUNI-Transformer	95.53	84.23	89.52	80.00	72.75	76.21	82.65	74.76	78.51
CUNI-T2T-2018	96.80	85.82	90.98	79.54	71.78	75.46	82.55	74.26	78.19
CUNI-Trf-T2T-2018	*96.76*	*84.75*	*90.36*	*79.85*	*71.71*	*75.56*	*82.77*	*74.01*	*78.15*
CUNI-Trf-T2T-2019	*95.60*	*85.66*	*90.36*	*79.58*	*71.57*	*75.36*	*82.38*	*74.04*	*77.99*
CUNI-DocTrf-T2T	*95.60*	*85.66*	*90.36*	*79.58*	*71.57*	*75.36*	*82.38*	*74.04*	*77.99*
CUNI-DocTransformer	97.19	85.51	90.98	79.06	71.08	74.86	82.23	73.65	77.70
eTranslation	95.20	85.61	90.15	76.13	70.15	73.02	79.48	72.92	76.06
OPPO	96.03	86.43	90.98	74.35	68.55	71.33	78.23	71.81	74.88
CUNI-DocTrf-Marian	*96.00*	*85.71*	*90.57*	*72.45*	*68.51*	*70.42*	*76.61*	*71.69*	*74.07*
UEDIN	*96.30*	*83.27*	*89.31*	*72.96*	*67.85*	*70.31*	*77.02*	*70.70*	*73.72*
UEDIN-CUNI	95.98	85.36	90.36	71.24	66.07	68.56	75.69	69.65	72.54
Online-A	95.49	83.51	89.10	69.89	67.28	68.56	74.34	70.33	72.28
Online-G	96.77	85.11	90.57	68.74	65.41	67.04	73.76	69.17	71.39
Online-Y	*97.57*	*84.86*	*90.77*	*61.57*	*63.73*	*62.63*	*67.93*	*68.03*	*67.98*
Online-Z	97.57	84.86	90.77	61.67	61.01	61.34	68.19	65.82	66.98
parfda	*95.02*	*75.27*	*84.00*	*68.16*	*58.44*	*62.93*	*72.85*	*61.57*	*66.74*
Online-B	**98.44**	**88.11**	**92.99**	57.50	59.80	58.63	65.12	65.74	65.43
Online-X	*95.70*	*87.81*	*91.59*	*57.35*	*58.89*	*58.11*	*64.54*	*64.83*	*64.68*
Online-A	*95.88*	*83.21*	*89.10*	*58.36*	*58.25*	*58.30*	*65.17*	*63.33*	*64.24*
Online-B	*97.93*	*83.16*	*89.94*	*57.02*	*57.24*	*57.13*	*64.46*	*62.63*	*63.53*
zlabs-nlp	95.55	84.59	89.73	47.21	47.68	47.45	56.61	55.65	56.13

Table 4: Results for English–Czech. WMT 2019 submissions are displayed in gray italics.

English–German	In-domain synsets			Out-of-domain synsets			All synsets		
Submission	Prec.	Recall	F1	Prec.	Recall	F1	Prec.	Recall	F1
Tohoku-AIP-NTT	83.17	**77.09**	80.01	55.53	**57.93**	**56.71**	73.82	**71.11**	**72.44**
Facebook_FAIR	**83.43**	*76.99*	**80.08**	**56.29**	*55.10*	*55.69*	**74.48**	*70.05*	*72.19*
Online-B	82.52	77.27	79.81	52.48	56.45	54.39	72.40	70.88	71.63
Microsoft-sentence-level	*83.18*	*77.14*	*80.05*	*52.81*	*51.92*	*52.36*	*73.31*	*69.27*	*71.23*
OPPO	81.81	76.48	79.05	52.58	55.23	53.87	72.01	69.89	70.93
Huoshan_Translate	82.05	77.16	79.53	50.24	53.32	51.73	71.50	69.89	70.68
eTranslation	81.99	75.36	78.53	51.44	52.77	52.09	71.82	68.38	70.05
Online-B	*83.37*	*74.78*	*78.85*	*51.92*	*50.66*	*51.28*	*73.04*	*67.30*	*70.05*
Microsoft-document-level	*81.76*	*75.68*	*78.60*	*47.21*	*48.11*	*47.65*	*70.54*	*67.29*	*68.88*
Online-Y	*81.29*	*75.30*	*78.18*	*46.37*	*48.21*	*47.27*	*69.87*	*67.12*	*68.47*
AFRL	81.82	73.96	77.69	45.73	45.33	45.53	70.16	65.28	67.63
Online-G	81.44	73.76	77.41	46.61	45.44	46.02	70.21	65.09	67.55
Online-G	*81.44*	*73.76*	*77.41*	*46.61*	*45.44*	*46.02*	*70.21*	*65.09*	*67.55*
Online-A	81.26	73.45	77.16	45.72	43.05	44.35	70.00	64.09	66.92
DFKI-NMT	*80.70*	*74.37*	*77.41*	*44.95*	*42.04*	*43.44*	*69.54*	*64.39*	*66.87*
PROMT_NMT	79.62	72.84	76.08	42.65	47.05	44.74	67.24	65.24	66.23
MLLP-UPV	*79.90*	*73.60*	*76.62*	*44.03*	*39.63*	*41.72*	*68.90*	*63.01*	*65.82*
LMU-CTX-TF-Single	*79.55*	*72.51*	*75.86*	*43.93*	*41.99*	*42.94*	*68.23*	*63.13*	*65.58*
UEDIN	78.55	75.47	76.98	37.42	39.56	38.46	65.61	64.90	65.25
NEU	*78.39*	*73.50*	*75.86*	*41.91*	*41.53*	*41.72*	*66.83*	*63.75*	*65.25*
eTranslation	*80.44*	*71.00*	*75.43*	*43.47*	*40.48*	*41.92*	*68.69*	*61.65*	*64.98*
MSRA.MADL	*80.53*	*71.97*	*76.01*	*41.79*	*35.63*	*38.46*	*68.88*	*60.67*	*64.51*
UCAM	*78.21*	*72.70*	*75.35*	*40.41*	*37.28*	*38.78*	*66.61*	*61.77*	*64.10*
Online-A	*79.21*	*72.05*	*75.46*	*40.48*	*36.44*	*38.35*	*67.37*	*61.09*	*64.07*
Helsinki-NLP	*78.34*	*72.52*	*75.32*	*39.06*	*36.65*	*37.82*	*66.24*	*61.57*	*63.82*
PROMT_NMT	*78.08*	*72.40*	*75.13*	*36.99*	*34.16*	*35.52*	*65.61*	*60.77*	*63.10*
Online-Z	75.61	69.71	72.54	41.06	43.03	42.02	64.18	61.62	62.87
JHU	*77.80*	*71.48*	*74.50*	*37.77*	*29.35*	*33.04*	*66.47*	*58.08*	*61.99*
UdS-DFKI	*78.27*	*70.54*	*74.21*	*35.68*	*30.16*	*32.69*	*65.72*	*58.10*	*61.68*
Online-X	*71.01*	*72.71*	*71.85*	*34.36*	*40.47*	*37.17*	*59.07*	*63.16*	*61.05*
zlabs-nlp	77.33	66.55	71.54	36.78	28.87	32.35	65.36	54.70	59.55
TartuNLP-c	*77.32*	*66.29*	*71.38*	*33.02*	*26.13*	*29.17*	*64.34*	*53.85*	*58.63*
WMTBiomedBaseline	73.59	57.02	64.25	31.91	15.52	20.88	63.33	42.82	51.09
EN_DE_Task	*64.54*	*23.14*	*34.06*	*38.41*	*5.64*	*9.84*	*59.43*	*16.62*	*25.97*

Table 5: Results for English–German. WMT 2019 submissions are displayed in gray italics.

German–English	In-domain synsets			Out-of-domain synsets			All synsets		
Submission	Prec.	Recall	F1	Prec.	Recall	F1	Prec.	Recall	F1
Online-G	80.35	**86.75**	**83.43**	51.37	**75.37**	**61.10**	72.78	**84.40**	**78.16**
Facebook_FAIR	**80.78**	*85.80*	*83.21*	**52.77**	*72.56*	**61.10**	**73.55**	*82.99*	*77.99*
Tohoku-AIP-NTT	80.52	86.32	83.32	48.56	72.84	58.27	72.21	83.62	77.50
OPPO	80.03	86.14	82.97	47.83	71.74	57.39	71.69	83.25	77.04
Online-B	80.36	83.75	82.02	48.79	69.68	57.39	72.16	80.88	76.27
Huoshan_Translate	78.11	86.00	81.86	45.05	71.06	55.14	69.53	83.06	75.70
Online-B	*77.88*	*83.81*	*80.73*	*45.50*	*66.51*	*54.04*	*69.58*	*80.31*	*74.56*
Online-G	*77.62*	*83.76*	*80.57*	*45.62*	*65.43*	*53.76*	*69.48*	*80.02*	*74.38*
Online-A	77.86	83.58	80.62	41.39	64.50	50.42	68.50	79.91	73.77
Online-Y	*76.82*	*84.51*	*80.48*	*41.93*	*61.71*	*49.93*	*68.10*	*79.97*	*73.56*
DFKI-NMT	*77.64*	*83.35*	*80.39*	*41.08*	*63.02*	*49.74*	*68.31*	*79.42*	*73.45*
RWTH_Aachen	*77.62*	*84.30*	*80.83*	*36.96*	*60.92*	*46.01*	*67.30*	*80.02*	*73.11*
MSRA.MADL	*77.95*	*84.36*	*81.03*	*36.73*	*56.26*	*44.44*	*67.78*	*79.08*	*73.00*
UCAM	*76.79*	*84.04*	*80.25*	*35.38*	*55.71*	*43.28*	*66.54*	*78.77*	*72.14*
MLLP-UPV	*77.26*	*83.24*	*80.14*	*35.85*	*54.92*	*43.38*	*67.02*	*77.93*	*72.06*
PROMT_NMT	75.14	83.75	79.21	38.74	60.85	47.34	65.95	79.33	72.02
Online-A	*75.77*	*83.08*	*79.26*	*37.47*	*63.15*	*47.04*	*65.87*	*79.40*	*72.00*
UEDIN	75.57	85.08	80.05	32.86	57.69	41.87	64.84	80.23	71.72
NEU	*75.26*	*83.50*	*79.16*	*32.49*	*55.93*	*41.11*	*64.49*	*78.58*	*70.84*
JHU	*74.94*	*83.68*	*79.07*	*31.56*	*51.38*	*39.10*	*64.31*	*77.79*	*70.41*
Online-Z	73.89	80.53	77.07	38.32	63.67	47.85	64.56	77.34	70.37
UEDIN	*74.26*	*81.62*	*77.77*	*32.21*	*45.89*	*37.85*	*64.28*	*74.70*	*69.10*
PROMT_NMT	*70.05*	*81.34*	*75.27*	*32.02*	*43.94*	*37.05*	*61.20*	*73.70*	*66.87*
Online-X	*67.04*	*80.29*	*73.07*	*31.98*	*62.47*	*42.31*	*57.77*	*77.07*	*66.04*
TartuNLP-c	*71.11*	*77.22*	*74.04*	*29.29*	*46.31*	*35.88*	*60.68*	*71.48*	*65.64*
WMTBiomedBaseline	69.23	70.34	69.78	23.05	22.63	22.84	59.54	60.05	59.79
zlabs-nlp	62.87	76.50	69.02	19.67	30.10	23.79	52.87	67.53	59.30

Table 6: Results for German–English. WMT 2019 submissions are displayed in gray italics.

English–Russian	In-domain synsets			Out-of-domain synsets			All synsets		
Submission	Prec.	Recall	F1	Prec.	Recall	F1	Prec.	Recall	F1
Online-G	**96.11**	89.64	92.76	**75.44**	74.52	**74.98**	**80.46**	78.35	**79.39**
Online-G	*95.56*	*89.58*	*92.47*	*75.11*	**74.85**	**74.98**	*80.05*	**78.58**	*79.31*
Facebook_FAIR	*95.49*	*88.28*	*91.75*	*67.68*	*71.54*	*69.56*	*74.40*	*76.01*	*75.20*
Online-B	94.97	89.01	91.89	63.86	71.67	67.54	71.35	76.44	73.81
OPPO	95.07	90.84	92.90	62.31	69.38	65.65	70.42	75.33	72.79
Online-B	*95.08*	*91.10*	*93.05*	*62.12*	*69.05*	*65.40*	*70.31*	*75.16*	*72.66*
USTC-MCC	*95.30*	*90.08*	*92.62*	*59.35*	*71.08*	*64.69*	*68.02*	*76.54*	*72.03*
NEU	*94.43*	*89.21*	*91.75*	*59.31*	*70.98*	*64.62*	*67.74*	*76.18*	*71.71*
Online-A	94.78	90.55	92.62	58.24	69.21	63.25	67.18	75.34	71.03
Ariel197197	95.66	85.97	90.56	61.40	66.77	63.97	69.70	72.12	70.89
Online-Y	*95.37*	*91.38*	**93.33**	*57.47*	*69.02*	*62.72*	*66.80*	*75.51*	*70.89*
PROMT_NMT	94.25	90.77	92.47	60.61	65.69	63.05	69.15	72.63	70.84
Online-A	*91.14*	*89.40*	*90.26*	*55.29*	*68.28*	*61.10*	*64.00*	*74.35*	*68.79*
PROMT_NMT	*93.48*	**91.49**	*92.47*	*56.78*	*63.76*	*60.07*	*66.18*	*71.61*	*68.79*
Online-X	*93.65*	*89.92*	*91.75*	*52.53*	*67.35*	*59.02*	*62.53*	*74.12*	*67.83*
Online-Z	95.80	88.83	92.18	53.95	60.97	57.24	64.56	69.13	66.76
zlabs-nlp	94.99	89.27	92.04	51.56	60.78	55.79	62.54	69.27	65.73
TartuNLP-u	*90.91*	*84.01*	*87.32*	*51.44*	*56.17*	*53.70*	*61.41*	*64.11*	*62.73*
Rerank-er	*94.98*	*78.91*	*86.20*	*55.54*	*33.78*	*42.01*	*68.17*	*45.36*	*54.47*
NICT	*89.19*	*25.52*	*39.68*	*46.99*	*5.88*	*10.46*	*63.90*	*10.33*	*17.78*

Table 7: Results for English–Russian. WMT 2019 submissions are displayed in gray italics.

References

Rohit Gupta, Constantin Orasan, and Josef van Genabith. 2015. ReVal: A simple and effective machine translation evaluation metric based on recurrent neural networks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1066–1072, Lisbon, Portugal. Association for Computational Linguistics.

Jenna Kanerva, Filip Ginter, and Tapio Salakoski. 2019. Universal lemmatizer: A sequence to sequence model for lemmatizing universal dependencies treebanks. *arXiv preprint arXiv:1902.00972*.

Massimiliano Mancini, Jose Camacho-Collados, Ignacio Iacobacci, and Roberto Navigli. 2017. Embedding words and senses together via joint knowledge-enhanced training. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 100–111, Vancouver, Canada. Association for Computational Linguistics.

Roberto Navigli and Simone Paolo Ponzetto. 2012. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250.

Alessandro Raganato, Yves Scherrer, and Jörg Tiedemann. 2019. The MuCoW test suite at WMT 2019: Automatically harvested multilingual contrastive word sense disambiguation test sets for machine translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 470–480, Florence, Italy. Association for Computational Linguistics.

Alessandro Raganato, Yves Scherrer, and Jörg Tiedemann. 2020. An evaluation benchmark for testing the word sense disambiguation capabilities of machine translation systems. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 3668–3675, Marseille, France. European Language Resources Association.

Hiroki Shimanaka, Tomoyuki Kajiwara, and Mamoru Komachi. 2018. RUSE: Regressor using sentence embeddings for automatic machine translation evaluation. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 751–758, Belgium, Brussels. Association for Computational Linguistics.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, pages 2214–2218, Istanbul, Turkey. European Language Resources Association (ELRA).

WMT20 Document-Level Markable Error Exploration

Vilém Zouhar **Tereza Vojtěchová** **Ondřej Bojar**

Charles University, Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics
Malostranské náměstí 25, 118 00 Prague, Czech Republic
{zouhar, vojtechova, bojar}@ufal.mff.cuni.cz

Abstract

Even though sentence-centric metrics are used widely in machine translation evaluation, document-level performance is at least equally important for professional usage. In this paper, we bring attention to detailed document-level evaluation focused on markables (expressions bearing most of the document meaning) and the negative impact of various markable error phenomena on the translation.

For an annotation experiment of two phases, we chose Czech and English documents translated by systems submitted to WMT20 News Translation Task. These documents are from the News, Audit and Lease domains. We show that the quality and also the kind of errors varies significantly among the domains. This systematic variance is in contrast to the automatic evaluation results.

We inspect which specific markables are problematic for MT systems and conclude with an analysis of the effect of markable error types on the MT performance measured by humans and automatic evaluation tools.

1 Introduction

This paper presents the results of our test suite for WMT20 News Translation Task.[1]

The conclusion of Vojtěchová et al. (2019), a last year's similar effort, states that expert knowledge is vital for correct and comprehensible translation of professional domains, such as Audits or Lease agreements. Furthermore, even MT systems which make fewer mistakes and score above others in both automatic and manual evaluations are prone to making fatal errors related to markable conflicts, which render the whole document translation unusable.

In this study, we aim to organize and describe a more detailed study with a higher number of annotators. We show three evaluation approaches: (1) automatic evaluation, (2) fluency and adequacy per document line and (3) detailed markable phenomena evaluation. We compare the results of this evaluation across the three domains and try to explain why all of these evaluations do not produce the same ordering of MT systems by performance.

This paper is organized accordingly: Section 1.1 defines the term "Markable", Section 1.2 describes the examined documents and Section 2 introduces the two phases of our annotation experiment and shows the annotator user interface in Section 2.3. In Section 3, we discuss the results from both phases and also automatic evaluation. The main results of this examination are shown in Section 3.5 and specific markable examples are discussed in Section 4. We conclude in Section 5.

1.1 Markable Definition

A markable in this context is an occurrence of any technical or non-technical term or expression that satisfies at least one of the following conditions:

1. The term was translated into two or more different ways *within one document*.
2. The term was translated into two or more different ways *across several translations*.
3. Two or more terms were translated to a specific expression *in one document* but have different meanings.

To be a markable, the term or expression does not have to be a named entity, but it must be vital to the understanding of the document. In the same order we show examples which satisfy the definition conditions.

1. *bytem* – It was translated within one document into *an apartment* and *a residence*.

[1] http://www.statmt.org/wmt20/translation-task.html

Proceedings of the 5th Conference on Machine Translation (WMT), pages 371–380
Online, November 19–20, 2020.

Document	Sentences	Direction	Markable occurrences	Description
Lease	29	cs→en en→cs	73 70	Housing lease agreement
Cars	18	cs→en	11	Brno Grand Prix competition article + highway accident report
Audit	90	cs→en en→cs	28 18	Supreme Audit Office audit report
Speech	13	en→cs	15	Greta Thunberg's U.N. speech article
Total	**269**	-	**215**	-

Table 1: Summary of examined documents with translation directions, number of lines and number of markable occurrences.

2. *rodné čísla* – It was translated in one translation to *social security number* and in another translation to *identification number*.
3. *nájemce*, *podnájemce* – They have different meanings and in one document they were both translated to tenant.

Markables were proposed first by the annotators in the first phase of annotation in Section 2.1 and then filtered manually by us.

1.2 Test Suite Composition

We selected 4 documents, 2 of which were translated in both directions totalling 6 documents. We chose 2 from the professional domain (Audit and Lease) and 2 from the News domain. The overview of their size is shown in Table 1. The number of markable occurrences is highly dependent on the document domain with the Agreement domain (Lease document) containing the most occurrences.

All of the MT systems are participants of the News Translation Task, and we test their performance even outside of this domain. Most of them were bi-directional, and we join the results from both directions when reporting their performance. The only exceptions are eTranslation (only en→cs) and PROMT_NMT (only cs→en).

1.3 Data and Tools Availability

All of the document translations and measured data are available in the project repository. Furthermore, the used online markable annotation tool written in TypeScript and Python is documented and also open-source.[2]

[2] github.com/ELITR/wmt20-elitr-testsuite

2 Annotation Setup

For both phases of this experiment, we used 10 native Czech annotators with English proficiency. None of them were professional audit or legal translators. Because each annotator annotated only one or two documents, the aggregated results across domains, labelled as *Total*, are of less significance than the results in individual domains.

2.1 Manual Document Evaluation

In this phase of the experiment, we wanted to measure the overall document translation quality and also to collect additional markables for use in the following experiment part. We showed the annotators the source document (in Czech) with a line highlighted and then underneath all its translation variants (in English). The current line was also highlighted. Next to every translation was a set of questions related to the just highlighted lines:

- **Adequacy**: range from 0 (worst) to 1 (best) measuring how much the translated message is content-wise correct regardless of grammatical and fluency errors.
- **Fluency**: range from 0 (worst) to 1 (best) measuring the fluency of the translation, regardless of the relation of the message to the source and the correct meaning.
- **Markables**: A text area for reporting markables for the second phase.
- **Conflicting markables**: checkbox for when there is a markable in conflict (e.g. the terminology change) with a previous occurrence in the document. This corresponds to the first condition in the markable definition in Section 1.1. The default value was *No* (no

conflict) because the distribution was highly imbalanced.

Bojar et al. (2016) summarize several methods for machine translation human evaluation: Fluency-Adequacy, Sentence Ranking, Sentence Comprehension, Direct Assessment, Constituent Rating and Constituent Judgement. For our purposes, we chose a method similar to Fluency-Adequacy as one of the standard sentence-centric methods. The difference to the method described is that we showed all the competing MT systems at once, together with the whole document context. Ultimately, we would like the users to rate Fluency-Adequacy of the whole documents, but we suspected that asking annotators to read the whole document and then rating it on two scales would yield unuseful biased results.

2.2 Manual Markable Evaluation

In the following phase, we focused on markables specifically. For every markable in the source, we asked the annotators to examine 11 phenomena. If the given phenomenon is present in the examined markable occurrence, a checkbox next to it should have been checked (Occurrence). Further on a scale 0–1 (not at all–most) the annotator should mark how negatively it affects the quality of the translation (Severity). We list the 11 phenomena we asked the annotators to work with:

- **Non-translated**: The markable or part of it was not translated.
- **Over-translated**: The markable was translated, but should not have been.
- **Terminology**: The translation terminology choice is terminologically misleading or erroneous.
- **Style**: An inappropriate translation style has been selected, such as too formal, colloquial, general.
- **Sense**: The meaning of the whole markable translation is different from what was intended by the source.
- **Typography**: Typographical errors in translation such as in capitalization, punctuation, special character or other typos.
- **Semantic role**: The markable has a different semantic role in translation than in the source. Without any specific linguistic theory in mind, we provided four basic roles for illustration: agent (story executor), patient (affected by the event), the addressee (recipient of the object in the event), effect (a consequence of the event).
- **Other grammar**: Other grammatical errors such as bad declension or ungrammatical form choice.
- **Inconsistency**: A different lexical translation option than the previous occurrence was used. It is enough to compare only with the previous occurrence and not with all of them.
- **Conflict**: The translation conflicts with another markable or term in the document. This and another markable translates to the same word.
- **Disappearance**: The markable does not appear in translation at all.

The choice to focus on markables was motivated by the aim to find a way to measure document-level performance using human annotators. A good markable translation is not a sufficient condition for document-level performance, but a necessary one. This approach is similar to Constituent Ranking/Judgement described by Bojar et al. (2016) with the difference that we chose to show all the markable occurrences in succession and in all translations in the same screen. We showed the whole translated documents context so that the annotators could refer to previous translations of the markable and the overall context.

2.3 Interface

Figure 1 shows the online interface for the second phase of this experiment. The first text area window contains the source document (e.g. in English). Below it are several translations (e.g. in Czech). Next to each translation is a set of questions. In the source, the current markable occurrence, to which the questions relate, is always displayed in dark blue. The current sentence is highlighted in the translations with light blue. The target words which probably correspond to the current markable (via automatic word alignment) are highlighted in dark blue as well. This alignment is present only for quick navigation as it is not very accurate. In translations, the remaining occurrences of a given markable are highlighted in green to simplify checking for inconsistency.

The FOCUS button is used to scroll to the current line in all text areas in case the user scrolled the view to examine the rest of the document.

In the first phase, the annotators could return to their previous answers and adjust them, but before

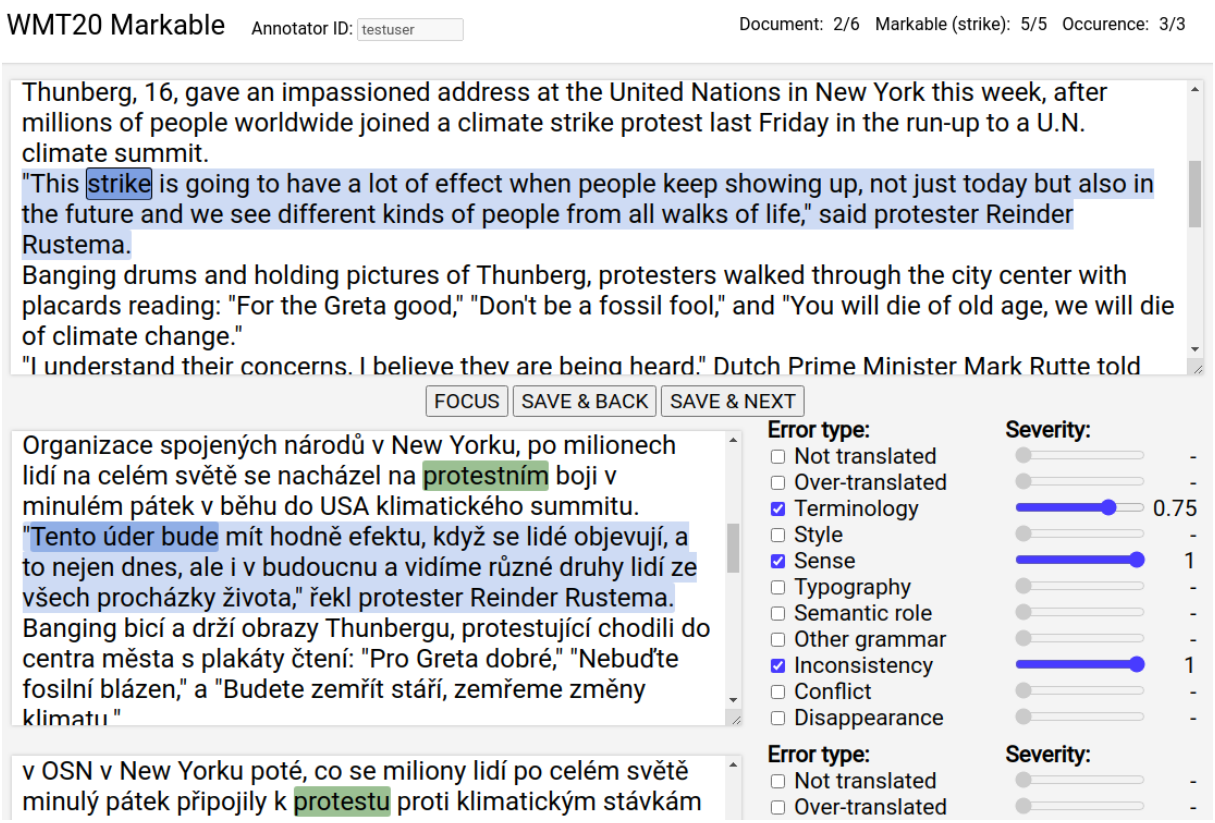


Figure 1: Online interface for markable annotation with highlighted segments. The 12 other translations are in the rest of the page, not fully visible here.

continuing to the next line, they had to fill in the current fluency and adequacy. In the second phase, the annotators could freely return to their previous answers and adjust them. The most straightforward approach for them was to annotate a single markable occurrence across all MT systems and the switch to the next one as opposed to annotating all markable occurrences in the first translation, then all markable occurrences in the second translation, and similarly the rest.

As soon as we aggregate the statistics over multiple documents (or even translation directions), the effects of which particular annotator annotated which document can start playing a role, but we hope they cancel out on average.

3 Results

3.1 Automatic Evaluation

We measured the system quality using BLEU (Papineni et al., 2002) against a single reference. The results sorted by the score across all documents are shown in Table 2. BLEU scores across different test sets are, of course, not comparable directly. Only a very big difference, such as that of eTranslation for News and Audit (39.43% and 23.23%) suggests some statistically sound phenomena. We measured the standard deviation across MT systems within individual domains: News (6.19), Audit (2.34) and News-Lease (2.74). The Audit domain was generally the least successful for most of the submitted systems (see Table 3) and the Lease domain was more stable in terms of variance. The MT system BLEU variance over annotated lines hints that the better the system, the higher variance it has. This may be because most of the best MT systems are focused on News and fail on other domains, while the lower performant MT systems are low performant systematically across all domains.

3.2 Overall Manual Evaluation

From the first phase (Section 2.1) we collected $13 \times 328 = 4264$ line annotations. From the second phase (Section 2.2) we collected $13 \times 499 = 6487$ markable annotations. The average duration for one annotation of one translated line in the first phase was 25s, while one annotation of one system-markable occurrence in the second phase took only 8s.

Fluency and Adequacy correlate per line together

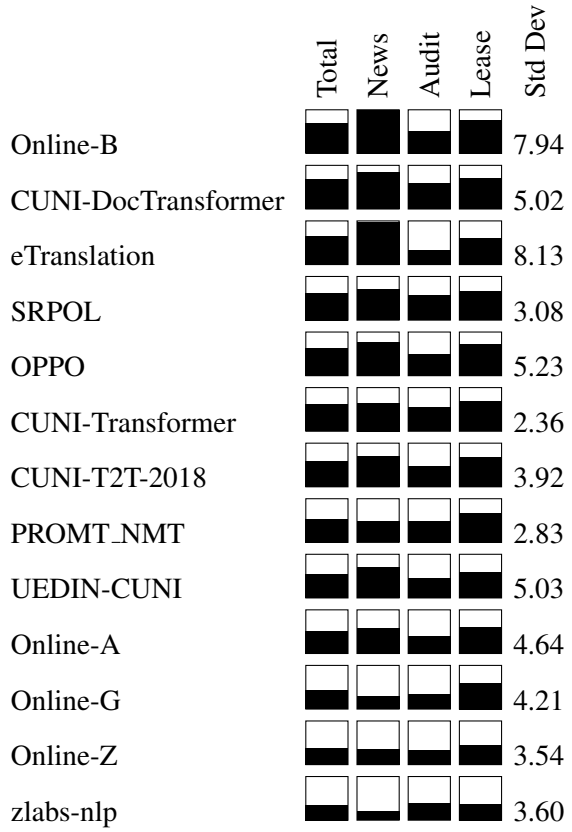


Table 2: MT system results measured by BLEU together with standard deviation measured from all sentences. Sorted by the first column. Full black box indicates 40% BLEU, empty 15% BLEU.

strongly (0.80), and their product correlates negatively (-0.33) with the number of wrong markables. Because of this strong correlation and also the need to describe the result of the first phase by one number, we focus on Fluency×Adequacy. Table 3 shows the average Fluency×Adequacy as well as the average number of reported wrong markables per line.

Document	Mult.	Mkbs.	BLEU
Audit →cs	0.95	0.08	28.61 ±5.13
Audit →en	0.81	1.23	32.68 ±5.07
Lease →cs	0.78	0.33	33.50 ±4.96
Lease →en	0.78	0.30	35.44 ±4.94
News →en	0.74	0.65	30.68 ±5.05
News →cs	0.65	0.83	38.67 ±4.93
Average	0.79	0.73	33.57 ±4.93

Table 3: Document average (across all systems) of Fluency×Adequacy (Mult.), number of reported wrong markables per line (Mkbs.) and BLEU.

3.3 MT System Performance

The performance per MT system and domain can be seen in Table 4. The reference translation received a comparably low rating in especially the Audit domain and fared best in the News domain. We see this as a confirmation of the last year's observation and a consequence of using non-expert annotators, who may have not annotated more complex cases thoroughly and were more content with rather general terms and language than what is correct for the specialized auditing domain.

No system has shown to be risky (high average but also with high variance). The last column in Table 4 shows, that the better the system, the more consistent it is (lower variation across documents). This did not occur with BLEU.

The ordering of systems by annotator assessment is slightly different than by automatic evaluation (Section 3.1). The automatic evaluation correlates with annotator rating (Fluency×Adequacy) with the coefficient of 0.93 (excluding Reference).

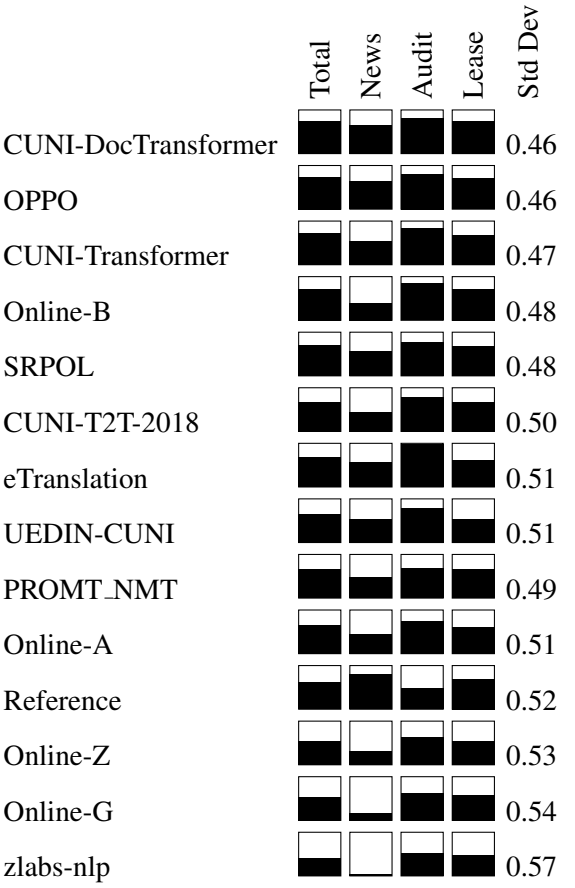


Table 4: MT system results measured by Fluency×Adequacy together with standard deviation measured from Total. Sorted by the first column. Full black box indicates 100%, empty 40%.

Notable is the distinction in the performance of eTranslation in the Audit domain. Its BLEU in this domain (23.23%, Table 2) was below average, however it performed best of all submitted MT systems in terms of Fluency×Adequacy (98.62%, Table 4), above Reference. Closer inspection revealed that the translations were very fluent and adequate but usually used vastly different phrasing than in the Reference, leading to very low BLEU scores.

Source:
In the vast majority of cases, the obligations arising from contracts for financing were properly implemented by the beneficiaries.
Reference:
Ve většině případů byly závazky vyplývající z podmínek podpory příjemci řádně plněny.
eTranslation: (**BLEU**: 9.24%)
Ve velké většině případů příjemci řádně plnili povinnosti vyplývající ze smluv o financování.
CUNI-DocTransformer: (**BLEU**: 41.21%)
V naprosté většině případů byly závazky vyplývající ze smluv o financování příjemci řádně plněny.

Figure 2: Example translations by eTranslation and CUNI-DocTransformer together with Source and Reference. N-grams present in Reference are underlined.

The example in Figure 2 shows activization (opposite of passivization) in the translation by eTranslation (*the beneficiaries fulfilled their obligations*) instead of (*obligations were fulfilled by the beneficiaries*). This resulted in much lower n-gram precision and BLEU score in general, even though the sentence is fluent and more adequate than both the Reference and translation by CUNI-DocTransformer.

3.4 Markable Phenomena and Systems

Table 5 shows an overview of types of markable phenomena with the average number of occurrences and Severity across systems. For all systems, *Terminology* and *Conflicting markables* had the most significant impact on the translation quality. These two categories clearly differ in Severity with markable conflicts being much more severe than terminological mistakes.

Inconsistency, *Typography* and *Disappearance* phenomena also heavily impacted the translation quality, although with varying distribution of Occurrences and Severity.

Reference differs from MT systems by having higher average Occurrence, but lower average Severity (first column in Table 5). Furthermore, the Reference had a higher number of *Inconsistence* occurrences, but with lower Severity. This means that most of these *Inconsitencies* were not actual errors. This is expected, as careful word choice variation improves the style and requires having an overview of previously used terms in the document.

Over-translation occurred rarely and in those cases, mostly in names (example shown in Figure 3). *Other grammar* manifested itself most severely in gender choice when translating sentences with person names without any gender indication from English to Czech. Similarly, *Style* was marked mostly in direct speech translation. The system used informal singular form addressing instead of plural. These two phenomena are shown in Figure 4.

Source & Reference: Karolína Černá
Translation: Caroline Black

Figure 3: Example of overly-translated named entity, it is the name of one of the parties in the Lease agreement.

Source:
“How dare you?” Thunberg’s U.N. speech inspires Dutch climate protesters
Reference:
“Jak se opovažujete?” projev Thunbergové v OSN inspiroval nizozemské protestující proti změnám klimatu
Translation:
“Jak se opovažuješ?” Thunbergův projev OSN inspiruje nizozemské klimatické demonstranty

---

Figure 4: Example of bad translation style.

Noteworthy is the correlation between phenomena across systems. The highest values were between *Sense* and *Terminology* (0.89), *Terminology* and *Inconsistency* (0.83) and *Sense* and *Other grammar* (0.82). There is no straightforward explanation of this correlation except the obvious that a good system is good across all phenomena. The correlation in the last phenomena pair suggests that the *Other grammar* category is too coarse and contains other subcategories.

3.5 Markable Phenomena and Domains

The results of markable phenomena across different domains is shown in Table 6.

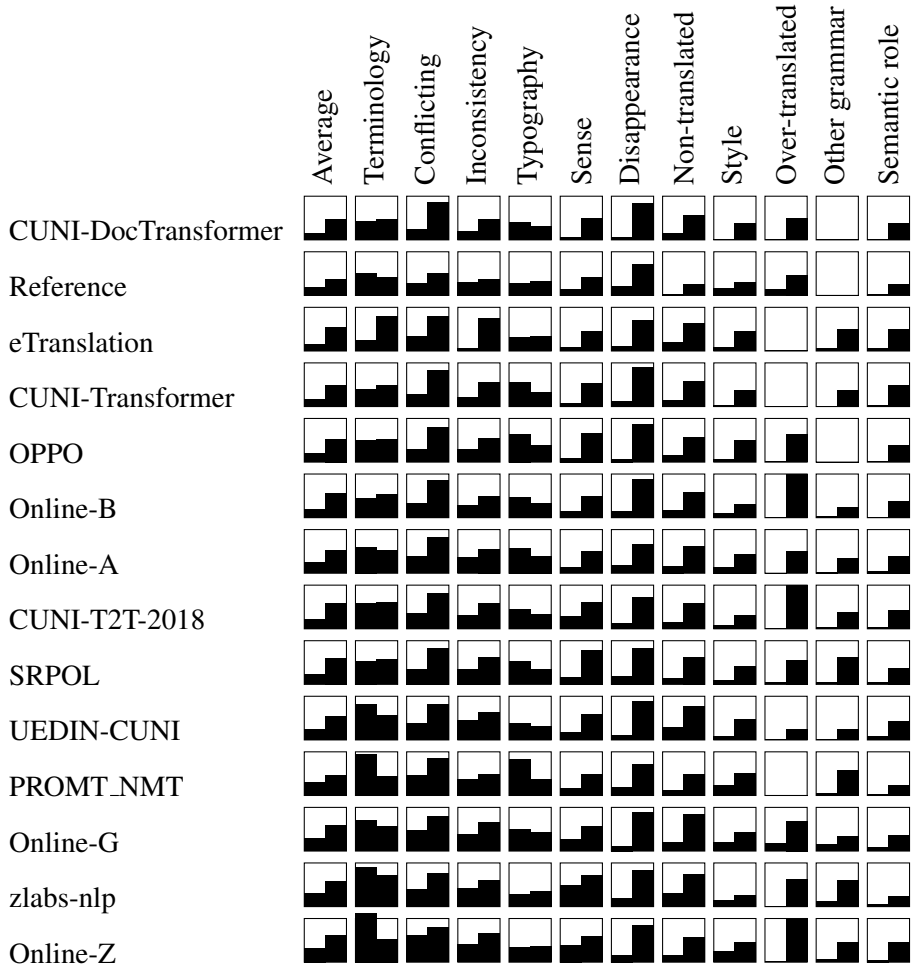


Table 5: Model results across 11 phenomena measured on markables together with their average. Each box is split into two bars: average Occurrence (left) and average Severity (right). Full left and right bars indicate occurrence in 20% of all markable instances and 100% Severity, respectively. Rows are sorted by Occurrence×Severity in the first column and columns, excluding *Average*, by the phenomena average Occurrence×Severity.

The second to last column is the correlation (across systems) between Occurrence×Severity and the BLEU score. The last column in Table 6 shows the correlation (across systems) between the two human scores: Occurrence×Severity and Fluency×Adequacy from the first phase of this experiment.

Since both BLEU and Fluency×Adequacy are positive metrics (the higher the score, the better the performance) and Occurrence×Severity is an error metric (the higher the number, the worse the performance), high negative correlations mean, that the metrics are mutually good performance predictors.

The strongest correlations are: *Conflicting* (-0.58), *Non-translated* (-0.55) and *Semantic role* (-0.41). Except for *Non-translated*, the reason is clear: BLEU is unable to check grammatical relations and never looks across sentences. We find the fact, that BLEU result was in agreement with error marking for these phenomena, to be positive.

Positive correlations (i.e. mismatches) were reached for *Disappearance* (0.28) and *Over-translated* (0.33), which is somewhat surprising because here BLEU *has* a chance to spot these errors from the technical point of view: shorter output could fire brevity penalty and missing terms where the exact wording is clear because they appear already in the source should decrease BLEU score. The overall correlation between Occurrence×Severity and Fluency×Adequacy is more significant than the correlation with BLEU. The most correlating variables are: *Sense* (-0.84), *Other grammar* (-0.84), *Terminology* (-0.81) and *Inconsistency* (-0.59).

Interesting is the markable phenomena *Disappearance* and *Sense* because of their high difference in correlations between BLEU and human score correlations.

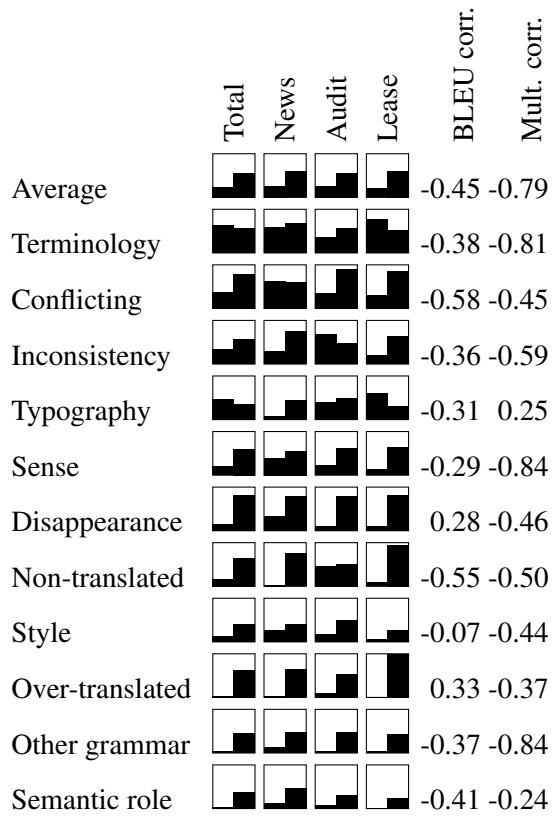


Table 6: Document domain average (across all systems) of markable phenomena. Sorted by Occurrence×Severity in the first column. Full left and right bars indicate occurrence in 20% of all markable instances and 100% Severity, respecively. The last two columns show correlation between Occurrence×Severity and BLEU and user ratings from Phase 1, respectively.

3.6 Annotator Agreement

We would like to bring attention to inter-annotator agreement for the second annotation phase. Table 7 lists the following metrics, which are computed pairwise and then averaged:

Plain inter-annotator agreement (IAA) reports the percentage of pairs of annotations where the two annotators agree that a given phenomenon was or was not present. IAA shows high numbers in all cases but it is skewed by the heavily imbalanced class distribution: most often, a phenomenon is not present; see the left sides of squares in the leftmost column in Table 6 for distribution reference.

Cohen's Kappa (Kappa), measured also pairwise, isolates the effect of agreeing by chance and reveals that a good agreement is actually reached only in the cases of *Disappearance*, *Terminology* and *Over-translated*, which are less ambiguous to annotate. It is unclear what is the reason behind the low Kappas, but we speculate that it is due to insufficient attention of the annotators: they would perhaps agree much more often that an error occurred but they were overloaded with the complexity of the annotation task and failed to notice on their own.

Phenomenon	IAA	Kappa	Corr.	Corr.+
Disappearance	0.90	0.43	0.52	0.06
Typography	0.95	0.20	0.55	-0.13
Sense	0.91	0.17	0.73	-0.09
Style	0.94	0.24	1.00	0.19
Terminology	0.90	0.41	0.07	-0.03
Inconsistency	0.88	0.13	0.18	-0.08
Non-translated	0.94	0.20	0.64	0.30
Conflicting	0.77	0.02	1.00	0.62
Other grammar	0.96	0.10	1.00	-0.35
Semantic role	0.97	-0.01	-	0.43
Over-translated	0.98	0.37	1.00	1.00

Table 7: Annotator agreement of Occurence marking (Inter Annotator Agreement and Cohen's Kappa) and agreement in Severity (two versions of Pearson Correlation) with respect to every markable phenomenon.

Plain Pearson Correlation (Corr.) was measured on Severities in instances where both annotators marked the phenomenon as present. This, however, disregards the disagreement in cases one annotator did not mark the phenomenon. For this, we also computed Corr.+, which examines all pairs in which at least one annotator reported Severity and replaces the other with zero.

We observe a big difference in the correlations. In cases where both annotators agreed that there was an error they tend to agree on the severity of the mistake, except *Terminology* and *Inconsistency*. If the cases where only one annotator marked the error are included, then the agreement on Severity is non-existent, except *Over-translation* and *Conflicting* translation.

3.7 Translation Direction

We also examined how the language translation directions affect the results. Most notable is CUNI-DocTransformer, which performs worse when translating into Czech. With only 0.01% higher Occurence of markable phenomena, the Severity increased by 20.81%. This is not something which we observed in other systems. The translation into Czech brought on average 0.01% higher Occurrence, but the Severity on average dropped by 3.99% when switching from English→Czech to Czech→English. The explanation supported by

the data is that in translation into English, CUNI-DocTransformer did not make any mistakes (or native Czech annotators did not detect them) and in translating into Czech, more issues were detected. Since the average Severity is measured across all phenomena, then the higher Severity in specific markable cases (Over-translated, Sense, Style and Disappearance) raised the overall average.

4 Annotation Examples

In the following figures (Figure 5, Figure 6 and Figure 7) we show annotated excerpts with BLEU, Fluency, Adequacy and markable phenomena severities. References are here to convey the Czech source segment meanings. They were not shown to the annotators. Examined markables are underlined.

Reference:
This Supplement No. 1 is written and signed in 2 (in words: two) copies, each of which is valid for the original.
Translation:
This Appendix 1 is drawn up and signed in two copies, each of which has the validity of the original.

BLEU: 23.59%, **Fluency:** 1, **Adequacy:** 0.9
Disappearance: 1

Figure 5: Example sentence markable (in words) annotation from Czech Lease document, translated by OPPO.

The example in Figure 5 focuses on intentional, key information duplication (for clarity and security reasons) of the number of signed copies. This duplication was however omitted in the translated output. The output is otherwise fluent and even received higher fluency than the Reference, which has an average fluency of 0.8.

Noteworthy is also another markable visible in the same figure, namely the referred section name: Appendix 1. Even though this word is different from the markable in the Reference: Supplement No. 1, it is used consistently across the whole document. Another variant of the translation is: Amendment No. 1. OPPO, together with Online-Z are the only systems which translated this markable correctly and consistently. Most of the systems (zlabs-nlp, Online-A, Online-B, Online-G, UEDIN-CUNI, CUNI-T2T-2018) switched inconsistently between the lexical choice. Other systems (SRPOL, eTranslation, CUNI-Transformer, CUNI-DocTransformer) were consistent in the main word choice, but not either in capitalization or number (e.g. Appendix No. 1 and Appendix 1).

Word variability (i.e. inconsistency) is often used to make the text more interesting, but in this context, it is vital that the term is translated consistently. Most of the systems, which outperformed even the Reference, made a severe error in this case.

Reference:
The most expensive item to be paid before the Grand Prix is the annual listing fee. This year, the fee was around 115 million Czech crowns. "Masses of people who come to Brno to see the Grand Prix spend money here for their accommodation, food and leisure activities, which should more or less balance out the cost associated with the organization of the event, including the listing fee," economist Petr Pelc evaluated the situation.
Translation:
The most expensive item is a breakdown fee every year before the Grand Prize. This year was about a hundred fifteen million crowns. "Mass of people who will come to Brno at the Grand Prix will spend money on accommodation, food or entertainment, which should more or less balance the costs associated with organizing the event, including the unifying fee," the economist Petr Pelc assessed.

BLEU: 26.59%, **Fluency:** 0.6, **Adequacy:** 0.4
Terminology: 1, **Sense:** 1, **Inconsistency:** 1

Figure 6: Example sentence markable (listing fee) annotation from Czech News document, translated by CUNI-T2T-2018.

Figure 6 shows a listing fee incorrectly translated as breakdown and unifying fee. This markable translation is interesting in the fact that systems were again very inconsistent with the markable translation choice. The wrong lexical choices were: landing, paving, parking, refill, landfill, security, zalistovacího, leasing, drop-in, back-up, reforestration, clearance, referral, padding fee and stamp duty. Good translations were: listing and registration fee.

Online-B and CUNI-DocTransformer made good and consistent lexical choices. SRPOL made good lexical choices but switched between them.

In this instance, this would not be an error, because consistency is not vital for interpreting the text.

The translation by CUNI-T2T-2018 in Figure 6 is not wrong only because of this markable translation choice, but also by poor fluency. The BLEU score, however, does not suggest, that there is anything fundamentally wrong with the translated segment despite the meaning being distorted.

Reference:
In Art. III of the Sublease agreement, entitled "Term of the Lease," the tenant and the lessee agreed that the apartment in question would be rented to the tenant for a fixed period from 13th May 2016 to 31st December 2018.
Translation:
In art. III of the apartment lease agreement, called "sublease period", the tenant and the tenant agreed that the apartment in question will be left to the tenant for use for a fixed period from 13. 5. 2016 to 31. 12. 2018.

BLEU: 31.95%, **Fluency:** 0.7, **Adequacy:** 0.5
Terminology: 0.5, **Sense:** 0.25, **Conflict:** 1,
Other grammar: 0.25

Figure 7: Example sentence markable (lessee) annotation from Czech News document, translated by Online-G.

The last example, in Figure 7, concerns itself with conflicting markables. In this case, two distinct markables (tenant and lessee) were merged into one translation tenant. This is a very fundamental error because, in the Lease agreement, these two markables refer to the two parties, which enter the contract.

Again, the BLEU does not suggest that anything is wrong with the translation. It could be even higher (51.06%) were it not for the localized date format in the Reference.

5 Conclusion

In this article, we compared three approaches to document translation evaluation. We saw that non-expert annotators rate most MT systems higher than Reference with Fluency and Adequacy, but Reference ranks better than most of them when inspecting markable phenomena and their Severity. Inspecting specific instances in detail, we found out that MT systems made errors in terms of markables, which no human translator would do.

Relating the current observation with the impression last year, we conclude that annotators lacking in-depth domain knowledge are not reliable for annotating on the rather broad scales of Fluency and Adequacy but they are capable of spotting term translation errors in the markable style of evaluation. This is important news because expert annotators can not be always secured. Unfortunately, the inter-annotator agreement remains generally low, possibly due to a high cognitive load with many systems annotated.

We further examined these markable phenomena and showed that especially *Sense*, *Other grammar* and *Terminology* kinds of errors negatively influence the Fluency and Adequacy the most. For BLEU the variables of highest importance were *Non-translated* and *Conflicting* errors.

In future work, we would like to examine more of the kinds of markable errors in modern MT systems and their influence on the translation quality. This description could then help researches focus on specific parts of their MT systems.

Furthermore, we would like to explore possible automated metrics, which would help in determining whether the document meaning remained intact with respect to markables.

Annotating markables appears to be easier for human annotators and more reliable for non-expert ones, and the results gave us more insight into the systems' performance than the Fluency-Adequacy method.

Acknowledgement

This study was supported in parts by the grants H2020-ICT-2018-2-825460 (ELITR) and Czech Science Foundation (grant n. 19-26934X, NEUREM3).

References

Ondřej Bojar, Christian Federmann, Barry Haddow, Philipp Koehn, Matt Post, and Lucia Specia. 2016. Ten years of WMT evaluation campaigns: Lessons learnt.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311–318, USA. Association for Computational Linguistics.

Tereza Vojtěchová, Michal Novák, Miloš Klouček, and Ondřej Bojar. 2019. SAO WMT19 test suite: Machine translation of audit reports. *arXiv preprint arXiv:1909.01701*.

Translating Similar Languages: Role of Mutual Intelligibility in Multilingual Transformers

Ife Adebara **El Moatez Billah Nagoudi** **Muhammad Abdul Mageed**
Natural Language Processing Lab
University of British Columbia
{ife.adebara,moatez.nagoudi,muhammad.mageed}@ubc.ca

Abstract

We investigate different approaches to translate between similar languages under low resource conditions, as part of our contribution to the WMT 2020 Similar Languages Translation Shared Task. We submitted Transformer-based bilingual and multilingual systems for all language pairs, in the two directions. We also leverage back-translation for one of the language pairs, acquiring an improvement of more than 3 BLEU points. We interpret our results in light of the degree of mutual intelligibility (based on Jaccard similarity) between each pair, finding a positive correlation between mutual intelligibility and model performance. Our Spanish-Catalan model has the best performance of all the five language pairs. Except for the case of Hindi-Marathi, our bilingual models achieve better performance than the multilingual models on all pairs.

1 Introduction

We present our findings from our participation in the WMT 2020 Similar Language Translation shared task, which focused on translation between similar language pairs in low-resource settings. Similar languages share a certain level of mutual intelligibility that may aid the improvement of translation quality. Depending on the level of closeness, certain languages may share similar orthography, lexical, syntactic, and or semantic structures which may make translation more accurate.

The level of mutual intelligibility is such that speakers of one language can understand another language without prior instruction in that other language. They can also communicate without the use of a lingua franca which is a link or vehicular language used for communicating between speakers of different languages (Gooskens, 2007). It is important to mention that, sometimes, the level of intelligibility varies in both directions. For instance, Slovene - Croatian intelligibility is said to be asymmetric such that speakers or Slovene can understand spoken and written Croatian better than speakers of Croatian understand Slovene (Golubović and Gooskens, 2015).

Machine translation of similar languages has been explored in a number of works (Hajic, 2000; Currey et al., 2016; Dabre et al., 2017). This can be seen as part of a growing need to develop models that translate well in low resource scenarios. The goal of the current shared task is to encourage researchers to explore methods for translating between similar languages. We also view the shared task as useful context for studying interaction between degrees of similarity and mutual intelligibility on the one hand, and model performance on the other hand. We explore the use of bilingual and multilingual models for all the 5 shared task language pairs. We also perform back-translation for one language pair.

In the remainder of this paper, we discuss related literature in Section 2. We explain the methodology which includes a description of the Transformer model, back-translation and beam search in Section 3. In Section 4, we describe the models we developed for this task and we discuss the various experiments we perform. We also describe the architectures of the models we developed. Then we discuss the evaluation procedure in Section 6. Evaluation is done on both the validation and test sets. We conclude with discussion and the insights gained from this task in Section 7.

2 Related Work

Translation between similar languages has recently attracted attention. Different approaches have been adopted using state-of-the-art techniques, methods, and tools to take advantage of the similarity between languages even in low resource scenarios. Approaches that have been effective for other ma-

Proceedings of the 5th Conference on Machine Translation (WMT), pages 381–386

chine translation tasks have proven to achieve success in the context of similar language translation as well.

NMT models, specifically the Transformer architecture, has been shown to perform well when translating between similar languages (Baquero-Arnal et al., 2019; Przystupa and Abdul-Mageed, 2019). The use of in-domain data for fine-tuning has also proven to be of remarkable benefit for this task. This problem has also been tackled both by using character replacement to leverage the orthographic and phonological relationship between closely related mutually intelligible language pairs (Chen and Avgustinova, 2019). A new approach was also introduced for this task using a two-dimensional method that assumes that each word of the target sentence can be explained by all the words in the source sentence (Baquero-Arnal et al., 2019).

Within the realm of MT for low resource languages, recent work has focused on translation using large monolingual corpora due to the scarcity of parallel data for many language pairs (Lample et al., 2018, 2017; Artetxe et al., 2018b). These approaches have leveraged careful initialization of the unsupervised neural MT model using an inferred bilingual dictionary, sequence-to-sequence language models, and back-translation to achieve remarkable results. The bilingual dictionary is built without parallel data by using an unsupervised approach to align the monolingual word embedding spaces from each language (Conneau et al., 2017; Artetxe et al., 2018a). Since parallel data is not available in sufficiently large quantities, back-translation is used to create pseudo parallel data. The monolingual data of the target language is translated into the source using an existing translation system (e.g., one trained with available gold data). The output is then used to train a new MT model (Sennrich et al., 2015a). Weak supervision caused by back-translation results in a noisy training dataset. This eventually can affect translation quality.

More recent works adopt different approaches to manage noise in back-translation. For instance, phrase based statistical MT models are introduced as a posterior regularization during the back-translation process to reduce the noise and errors of the data generated (Ren et al., 2019). Another method (Artetxe et al., 2019b) uses cross lingual word embeddings incorporated with sub-word information. The weights of the log-linear model is then tuned through an unsupervised process and the entire system is jointly refined in opposite directions to improve performance. This method outperforms previous SOTA model with about 5-7 BLUE points. A re-scoring mechanism that re-uses the pre-trained language model to select translations generated through beam search has also been found to improve fluency and consistency of translations (Liu et al., 2019). Yet another approach, combines cross-lingual embeddings with a language model to make a phrase-table (Artetxe et al., 2019a). The resulting system is then used to generate a pseudo parallel corpus with which a bilingual lexicon is derived. This approach can work with any word or cross-lingual embeddings techniques.

3 Methodology

Motivated by the success of Transformers and back-translation, we develope a sequence-to-sequence approach using the Transformer architecture perform back-translation for one language pair. For decoding, we use Beam Search (BS). BS is an heuristic decoding strategy based on exploring the solution space and selecting a sequence of words that maximize the overall likelihood of the target sentence. During the translation, we hold a beam of β sequences (*beam size*) which are iteratively extended. At each step, β words are selected to extend each of the sequences in the beam, so the output is β^2 candidate sequences (hypotheses), we retain only the β highest score hypotheses for the next step (top-β candidates) (Koehn, 2009). In all our experiments we use beam size of 5 whilst decoding.

3.1 Transformer

Our baseline models are based on the Transformer architecture. A Transformer (Vaswani et al., 2017) is a sequence-to-sequence model that does not have the recurrent architecture present in Recurrent Neural Networks (RNNs). It uses a positional encoding that can remember how sequences are fed into the model. These positions are added to the embedded representation (n-dimensional vector) of each word. Transformers have been shown to train faster than RNNs for translation tasks.

The encoder and decoder in a Transformer model have modules that consist mainly of multi-head attention and feedforward layers. The attention mechanism is based on a function that operates on Q (*queries*), K (*keys*), and V (*values*). The query

is a vector representation of one token in the input sequence, K refers to the vector representations of all the tokens in the input sequence. More information about the Transformer are in (Vaswani et al., 2017).

3.2 Back-translation

We perform back-translation using the monolingual model developed for the Croatian-Slovene (HR-SL) language pair. We use the best HR-SL model checkpoint that acquire the highest BLEU score on the DEV set to translate the monolingual HR data. This produces synthetic Slovene (SL) data which we then use as the source language while the original monolingual data is used as target when training the SL-HR model. We combine this data with the initial training data.

Due to time constraints, we used only a subset of the monolingual data with a beam size of 5.

3.3 Jaccard Similarity

Jaccard similarity compares similarity, diversity, and distance of data sets (Niwattanakul et al., 2013). It is calculated between two data sets in our case two languages) A and B by dividing the number of features common to the two sets (their intersection) by the union of features in the two sets, as in (1) below:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} \quad (1)$$

We use tokens, identified based on white space, as features when we calculate Jaccard.

4 Experiments

4.1 Model Architecture

Our neural network models are based on the Transformer architecture (as described in Section 3) implemented by Facebook in the Fairseq toolkit. The following hyper-parameter configuration was used: 6 attention layers in the encoder and the decoder, 4 attention heads in each layer, embedding dimension of 512, maximum number of tokens per batch was set to $4,096$, Adam optimizer with $\beta 1 = 0.90$, $\beta 2 = 0.98$, dropout regularization was set to 0.3, weight-decay was set at 0.0001, $label - smoothing = 0.1$, variable learning rate set at $5e-4$ with the inverse square root, lr-scheduler and $warmup - updates = 4,000$ steps. We used the label smoothed cross-entropy criterion, and gradient clip-norm threshold was set to 0.

5 Data

We used all the parallel data for all language pairs http://www.statmt.org/wmt20/similar.html. The task was constrained so we did not add any additional data to develop our models. We used the monolingual data for the SL-HR language pair for back-translation. Table 2 shows the size of the data in terms of the number of sentences and words for each language pair while Table 1 shows example source and corresponding outputs from our bilingual and multilingual models for each language pair. We also calculated the jaccard similarity for the training data we used for the tasks. "Jaccard similarity" measures the similarity between two text documents by taking the intersection of both and dividing it by their union. Linguists measure these intersections (Oktavia, 2019; Gooskens and Swarte, 2017) between languages to determine the level of mutual intelligibility as well as classify languages as dialects of the same language or different languages. We calculated Jaccard similarity for each language pair.

5.1 Pre-processing

Pre-processing was by a regular Moses toolkit (Koehn et al., 2007) pipeline that involved tokenization, byte pair encoding and removing long sentences. We applied Byte-Pair Encoding (BPE) (Sennrich et al., 2015b) operations, learned jointly over the source and target languages. For each language pair, we used 32k split operation for subword segmentation (Sennrich et al., 2016b). We run experiments with Transformers under three settings, as we explain next.

5.2 Models

We develop both bilingual and multilingual models using gold data for all pairs. For one pair, we also use back-translation with one bilingual model. We provide more details next.

5.2.1 Bilingual Models

In this setting, we build an independent model for each language pair. We develop models for both directions for all language pairs, thus ultimately creating 12 models (6 for each direction). We train each model on 1 GPU for 7 days.

5.2.2 Multilingual Models

We develop two multilingual models that translates between all languages; a model for each direction (2 models overall) (Johnson et al., 2017). We add

Model	Pair	Sentence	Translation
Bilingual Model	Es-Ca	Diseña stickers para soñar	Dissenya stickers per soyir
		Mueva el diez de corazones al nueve de corazones .	Mobles el deu de garrons al nou de garrons .
	Es-Pt	Diseña stickers para soñar	Design stickers para sonhar
		Mueva el diez de corazones al nueve de corazones .	Muda o dez corações para nove corações .
	Sl-Hr	Vesel pomladni pozdrav ob novi izdaji Bisnode novičk .	Sretan proljetni pozdrav na novom izdanju Bisnode Vijesti .
		Z lepimi pozdravi ,	S lijepim pozdravima ,
	Sl-Sr	Danes ni enostavno slediti vsem informacijam , ki so pomembne za poslovanje .	Danas nije lako pratiti sve informacije koje su važne za poslovanje .
		Iščete podatke za drugo državo ?	Tražite podatke za drugu zemlju ?
Multilingual Model	Es-Ca	Mueva el diez de corazones al nueve de corazones .	Muva el 10 de coração al 9 de coraons .
		el cuatro de diamantes	el quatre de diamants
	Es-Pt	Luche en el aire con un avión enemigo	Luche en l aire amb un avió enemigo
		Entonces , ¿ qué salió mal ?	Então , o que saiu mal ?
	Sl-Hr	Objašnjenje - Indeks plaćanja Datum :	Objašnjenje – Indeks plaćanja Datum :
		Poštovani ,	Poštovani ,
	Sl-Sr	Iščete podatke za drugo državo ?	Tražite podatke za drugu zemlju ?
		Vesel pomladni pozdrav ob novi izdaji Bisnode novičk .	Sretan proljetni pozdrav uz novo izdanje Bisnode novosti .

Table 1: Examples sentences from the various pairs and corresponding translations based on the bilingual and multilingual models. Examples are from the DEV set.

	Language	#sentences	#words
Hi-Mr	hi	$43.2K$	$829.9K$
	mr	$43.2K$	$600K$
	mono-hi	$113.5M$	$4.74B$
	mono-mr	$4.9M$	$112.6M$
Es-Ca	es	$11.3M$	$150.4M$
	ca	$11.3M$	$163M$
	mono-es	$58.4M$	$1.5B$
	mono-ca	$28M$	$763.7M$
Es-pt	es	$4.15M$	$86.6M$
	pt	$4.15M$	$82.5M$
	mono-es	$58.4M$	$1.47B$
	mono-pt	$11.4M$	$233.9M$
Sl-Hr	sl	$17.6M$	$113.09M$
	hr	$17.6M$	$117.73M$
	mono-sl	$46.25M$	$770.6M$
	mono-hr	$64.5M$	$1.24B$
Sl-Sr	sl	$14.1M$	$79.1M$
	sr	$14.1M$	$86.1M$
	mono-sl	$46.2M$	$770.6M$
	mono-sr	$24M$	$489.9M$

Table 2: Number of sentences and words for the training data used for each language pair

a language code representing the target language as the start token for each line of the source data. We train each model on 4 GPUs for 7 days. We use the same hyper-parameters values set for the bilingual models. Multilingual models enable us to determine the impact of learning similar languages with a shared representation.

5.2.3 Bilingual Model with Back-translation

For the third approach, we combine back-translation with the bilingual translation model for the SL-HR language pair. We incorporated the monolingual data to do this. This was influenced by report (Sennrich et al., 2016a) in literature on the significant improvement of translation quality when monolingual data is incorporated into training data through back-translation. We were able to test the effect of back-translation on one model.

6 Evaluation

We evaluated both the DEV and TEST sets. We used the best-checkpoint metric with BLEU score to evaluate the validation set at each iteration. We used a beam size of 5 during the evaluation . We de-tokenized from BPEs back into words.

6.1 Evaluation on DEV set

We report the results on the DEV sets for each language pair in Table 3. These models were trained without the monolingual data except the SL-HR pair with the asteriks.

Pair	Bilingual Models	Multiling. models
hi-mr	12.14	16.35
mr-hi	10.63	01.02
es-ca	74.85	16.13
ca-es	74.24	64.57
es-pt	46.71	26.41
pt-es	41.12	05.81
∗sl-hr	36.89	-
sl-hr	33.28	09.25
hr-sl	55.51	07.94
sl-sr	40.80	32.80
sr-sl	39.80	06.97

Table 3: Evaluation in BLEU on the development set for the different language pairs. The asteriks shows the model with back-translation

The bilingual models outperform the multilingual models for all language pairs except the hi-mr language pair.

6.2 Evaluation on TEST

In order to evaluate the test data, we removed the byte-pair code from the test set. We used the fairseq-generate mode while translating the test set. We show results on the test set in Table 4.

Pair	Bilingual Models	Multiling. models
hi-mr	0.49	-
es-ca	41.74	8.49
ca-es	45.23	45.86
es-pt	23.35	17.06
pt-es	24.26	21.55
*sl-hr	20.92	22.26
hr-sl	14.94	7.37
sl-sr	14.7	20.18
sr-sl	19.46	11.37

Table 4: The BLEU scores for some of the models on the test set. The language pair with the asterisks has back-translation [1]

6.3 Discussion

We used the Jaccard similarity to measure the level of mutual intelligbility. Figure 1 shows a positive correlation between the BLEU scores and the Jaccard similarity between each language pair. [2] This relationship hold both for the bilingual and multilingual models. One exception is the Slovene-Serbian pair (SL-SR) where higher similarity does not translate into higher BLEU. For example, the SL-SR BLEU is below the SL-HR BLEU even though the latter pair has a higher similarity score. Interpreting Jaccard to mean mutual intelligibility, our findings imply a higher intelligibility is correlated with higher BLEU scores. However, there is a need to further investigate this relationship due to the SL-SR we observe.

7 Conclusion

We described our contribution to the WMT2020 Similar Languages Translation Shared Task. We developed both bilingual and multilingual models for all pairs, in both directions. We showed back-translation to help improve performance on one pair. We also showed how mutual intelligibility between a pair of languages (measured by Jaccard similarity) positively correlate with model performance (in BLEU). Future work can focus on exploiting other similarity metrics and providing a more in-depth study of mutual intelligibility between similar languages and how it interacts with MT model performance both in bilingual and multilingual models. The utility of back-translation on pairs we have not studied can also be fruitful.

[2]We multiplied the jaccard similarity by 100 to reduce the range of values on the y axis.

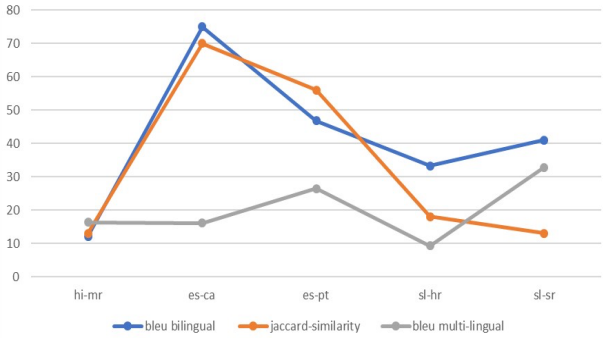


Figure 1: Interaction between performance in BLEU and Jaccard similarity.

Acknowledgements

MAM gratefully acknowledges the support of the Natural Sciences and Engineering Research Council of Canada (NSERC), the Social Sciences Research Council of Canada (SSHRC), and Compute Canada (www.computecanada.ca).

References

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018a. Generalizing and improving bilingual word embedding mappings with a multi-step framework of linear transformations. In Thirty-Second AAAI Conference on Artificial Intelligence.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2019a. Bilingual lexicon induction through unsupervised machine translation. arXiv preprint arXiv:1907.10761.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2019b. An effective approach to unsupervised machine translation. arXiv preprint arXiv:1902.01313.

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. 2018b. Unsupervised neural machine translation. arXiv preprint arXiv:1710.11041.

Pau Baquero-Arnal, Javier Iranzo-Sánchez, Jorge Civera, and Alfons Juan. 2019. The mllp-upv spanish-portuguese and portuguese-spanish machine translation systems for wmt19 similar language translation task. In Proceedings of the 4th Conference on MT (Volume 3: Shared Task Papers, Day 2), pages 179–184.

Yu Chen and Tania Avgustinova. 2019. Machine translation from an intercomprehension perspective. In Proceedings of the 4th Conference on MT (Volume 3: Shared Task Papers, Day 2), pages 192–196.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2017. Word translation without parallel data. arXiv preprint arXiv:1710.04087.

Anna Currey, Alina Karakanta, and Jon Dehdari. 2016. Using related languages to enhance statistical language models. In Proceedings of the NAACL Student Research Workshop, pages 116–123.

Raj Dabre, Tetsuji Nakagawa, and Hideto Kazawa. 2017. An empirical study of language relatedness for transfer learning in neural machine translation. In Proceedings of the 31st Pacific Asia Conference on Language, Information and Computation, pages 282–286.

Jelena Golubović and Charlotte Gooskens. 2015. Mutual intelligibility between west and south slavic languages. Russian Linguistics, 39(3):351–373.

Charlotte Gooskens. 2007. The contribution of linguistic factors to the intelligibility of closely related languages. Journal of Multilingual and multicultural development, 28(6):445–467.

Charlotte Gooskens and Femke Swarte. 2017. Linguistic and extra-linguistic predictors of mutual intelligibility between germanic languages. Nordic Journal of Linguistics, 40(2):123–147.

Jan Hajic. 2000. Machine translation of very close languages. In 6th Applied NLP Conference, pages 7–12.

Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. Transactions of the Association for Computational Linguistics, 5:339–351.

Philipp Koehn. 2009. Statistical machine translation. Cambridge University Press.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In Proceedings of the 45th annual meeting of the ACL companion volume proceedings of the demo and poster sessions, pages 177–180.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2017. Unsupervised machine translation using monolingual corpora only. arXiv preprint arXiv:1711.00043.

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Phrase-based & neural unsupervised machine translation. arXiv preprint arXiv:1804.07755.

Zihan Liu, Yan Xu, Genta Indra Winata, and Pascale Fung. 2019. Incorporating word and subword units in unsupervised machine translation using language model rescoring. arXiv preprint arXiv:1908.05925.

Suphakit Niwattanakul, Jatsada Singthongchai, Ekkachai Naenudorn, and Supachanun Wanapu. 2013. Using of jaccard coefficient for keywords similarity. In Proceedings of the international multiconference of engineers and computer scientists, volume 1, pages 380–384.

Diana Oktavia. 2019. Understanding new language: Mutual intelligibility in romance language pair. Journal Of Language Education and Development (JLed), 2(1):180–187.

Michael Przystupa and Muhammad Abdul-Mageed. 2019. Neural machine translation of low-resource and similar languages with backtranslation. In Proceedings of the 4th Conference on MT (Volume 3: Shared Task Papers, Day 2), pages 224–235.

Shuo Ren, Zhirui Zhang, Shujie Liu, Ming Zhou, and Shuai Ma. 2019. Unsupervised neural machine translation with smt as posterior regularization. arXiv preprint arXiv:1901.04112.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015a. Improving neural machine translation models with monolingual data. arXiv preprint arXiv:1511.06709.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015b. Neural machine translation of rare words with subword units. arXiv preprint arXiv:1508.07909.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In Advances in neural information processing systems, pages 5998–6008.

Attention Transformer Model for Translation of Similar Languages

Farhan
National University of Computer and Emerging Sciences, Karachi Campus
k180900@nu.edu.pk

Muhammad Rafi
National University of Computer and Emerging Sciences, Karachi Campus
muhammad.rafi@nu.edu.pk

Abstract

This paper illustrates our approach to the shared task on similar language translation in the fifth conference on machine translation (WMT-20). Our motivation comes from the latest state of the art neural machine translation in which Transformers and Recurrent Attention models are effectively used. A typical sequence-sequence architecture consists of an encoder and a decoder Recurrent Neural Network (RNN). The encoder recursively processes a source sequence and reduces it into a fixed-length vector (context), and the decoder generates a target sequence, token by token, conditioned on the same context. In contrast, the advantage of transformers is to reduce the training time by offering a higher degree of parallelism at the cost of freedom for sequential order. With the introduction of Recurrent Attention, it allows the decoder to focus effectively on order of the source sequence at different decoding steps. In our approach, we have combined the recurrence based layered encoder-decoder model with the Transformer model. Our Attention Transformer model enjoys the benefits of both Recurrent Attention and Transformer to quickly learn the most probable sequence for decoding in the target language. The architecture is especially suited for similar languages (languages coming from the same family). We have submitted our system for both Indo-Aryan Language forward (Hindi to Marathi) and reverse (Marathi to Hindi) pair. Our system trains on the parallel corpus of the training dataset provided by the organizers and achieved an average BLEU point of 3.68 with 97.64 TER score for the Hindi-Marathi, along with 9.02 BLEU point and 88.6 TER score for Marathi-Hindi testing set.

1 Introduction

This paper focuses on establishing a neural machine translation model using Encoder-Decoder architecture to translate between Hindi-Marathi sentence pairs. We are utilizing an attention mechanism approach by employing the combination of recurrence based RNN Encoder-Decoder layers (Cho and et al., 2014) and latest machine translation technique the Transformers (Vaswani et al., 2017) to address the complexity of WMT-20 similar language data-set (Hindi-Marathi). The motivation behind combining RNN and Transformer architecture is taken from the paper (Huang et al., 2020) to utilize the benefits of both. The formulated machine translation system has numerous applications. It can be used in advertising campaigns to reach native users. The media industry can employ this technology for generating subtitles and broadcasting multilingual news to cover a wide range of native subscribers of a region. Moreover, these systems can provide native vernaculars in social media to increase the activities of indigenous individuals of a native language. Additionally, Search engines can also adopt this method to display relevant results to clients in their region-specific native language.

The paper is composed of three further sections. The second section will display our proposed approach to solve the problem of Hindi-Marathi translation, followed by a discussion on designed experiments for training the model and mechanism for generating the results. Lastly, we will illustrate contributions and future work.

2 Methodology

The paper proposes a novel approach called Attention Transformer shown in Fig. 1 to translate between Hindi and Marathi text. In the figure, two encoder and decoder layers are illustrated. The initial N Encoder-Decoder layers contain RNN units in the form of LSTMs, which get stacked on top of each other. First, the initial encoder layer processes the tokenized input and generates a context vec-

Proceedings of the 5th Conference on Machine Translation (WMT), pages 387–392

tor. The initial decoder layer consumes the context vector generated by the neighboring encoder layer. Plus, it takes the target output as its input while training by utilizing the concept of teacher force training (Goyal and et al., 2016). Next, the processed outputs of both the initial Encoder-Decoder layer gets presented as input to the transformer model.

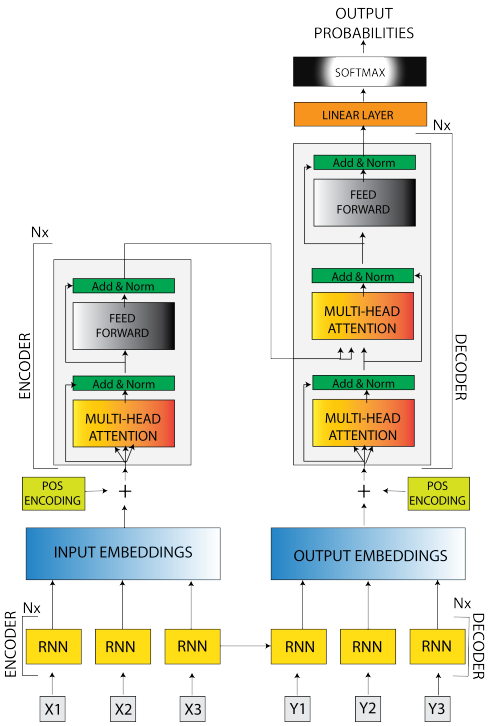


Figure 1: This Figure shows bird's-eye view of the proposed Attention Transformer Model

In the Transformer model, we can observe the second Encoder and Decoder layer stacked on the top of positional input-output embedding layers. In the Transformer, the positional input-output embedding layer receives the output of the initial Encoder-Decoder layer as its input. The role of the positional embedding layer is first to convert its input, which comes from the previous RNN based Encoder-Decoder layer into d-dimensional space where d is the output size of the embedding layer and add a positional encoding vector to it. As a result, all similar words relative to their positions in the training sentences will get cluster together. The working of the positional encoding vector is presented in the paper (Vaswani et al., 2017). Next, the outputs of the positional embedding layer get passed to Encoder-Decoder layers of Transformer. An individual Encoder-Decoder layer of Transformer contains a multi-head attention mechanism followed by a feed-forward layer, and a Transformer can have N number of such layers. The multi-head attention mechanism and feed-forward layer works, as illustrated in the paper (Vaswani et al., 2017), and their outputs are normalized. To train the model, we can apply the teacher forcing mechanism while during testing, a start token is initially required for the decoder to start the decoding process. After that, we can let the decoder to generate the output tokens in a loop by utilizing its current output as input to the next time frame until it produces the end token.

3 Experimental Studies

We have used Two human-level translation evaluation criteria, which are BLEU (Papineni and et al., 2002) and TER (Snover and et al.) scores and two general evaluation metrics that are Sparse Categorical Accuracy and Mean Loss. This section will first discuss the preparation of training data-set and baseline models, followed by training procedures plus their outcomes. And then, we will move towards explaining the results of the testing procedure. It is important to note that all experiments given below are performed using TPU with 180 TFlops, and 64 GB High Bandwidth Memory (HBM) provided by Google Colab, plus an implementation of the experiment is located in the Colab notebook (implementation).

3.1 *Data-set*

Initially, we have a Hindi-Marathi parallel training corpus of 44,685 sentence pairs. We have applied a simple rule to filter out all sentence pairs having the length higher than 24 words. After using this filtering rule on the data-set, we are left with 35,215 sentence pairs on which we have applied 80%-20% split to extract out training and development data. We have separated 100 records from the dev-set, and treat it as unseen data to perform a comparison with the baseline models. The table 1 shows the division of the data-set.

The reason for the maximum length based filtration of the data-set is with an increase in the maximum length of a sentence in the data-set, the complexity of the model increases, hence the training time increases. Although vocabulary size and hyper-parameters of the model also play a significant role in the training time per Epoch. Plus, we are motivated to keep our model simple as much as possible because the quality of the predicted translation gets affected, and it becomes difficult to

DATA-SET CONTENTS	
Data-set total sentence pairs	44,685
Filtered data-set sentence pairs	35,215
Training set sentence pairs	28,172
Development set sentence pairs	6,943
Records for comparison with baseline models	100
Hindi vocabulary size in filtered data	31,417
Marathi vocabulary size in filtered data	53,639

Table 1: The table shows the division of the data-set

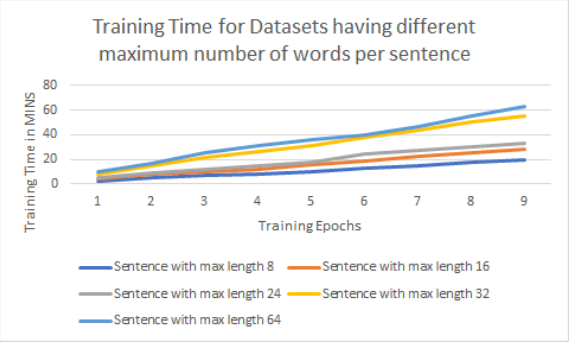


Figure 2: The figure shows line graphs illustrating the training time on filtered data-sets having different lengths of maximum words in a sentence.

debug the model as it grows more complex.

3.2 *Baseline Models*

We have selected the Bahdanau (Bahdanau and et al., 2014) and Transformer (Vaswani et al., 2017) model as a baseline model to compare the performance of our model. We have used their TensorFlow implementation officially given at (TensorFlow, a) and (TensorFlow, b). In our experiment, we have extracted 100 records from the dev-data, which serves as unseen data and helps us to compare the goodness of our proposed model with the selected baseline models. We have trained the model on the Marathi-Hindi dataset, with the mentioned parameters in TensorFlow documentation, and recorded that the Bahadanau model gets an average BLEU score of 0.13, while the transformer gets an average BLEU score of 20.

3.3 *Selecting Hyper-parameters*

The first essential hyper-parameter is to decide the maximum number of words a source or target sentence can have in a single given instance of a training sentence. However, it's a fantasy to develop a model that handles infinite words in the training instance. But as a result, it leads to infinite training time, which is undesirable. We have run the attention transformer model with the top 1000 records after filtering the dev-set with the various maximum number of words a source and target sentence can have and recorded their training time as shown in the chart below. In Fig 2, we can notice that increment in the maximum number of words a sentence can have produces a drastic impact on training time. We have selected 24 as the maximum number of words a sentence can have in our data-set to achieve comparable performance in practical training time.

After filtering the data-set based on the maximum number of words, a sentence can have. The next essential thing is to choose an appropriate batch size for the model. We have executed the Attention transformer model with different batch sizes on dev-set and noted their impact on training time per epoch. The Fig. 3 illustrates that the increment in batch size helps to reduce training time up to an extent after that it reduces the efficiency of the model. We have selected 16 as batch size, as it gives minimum training time per epoch for the model.

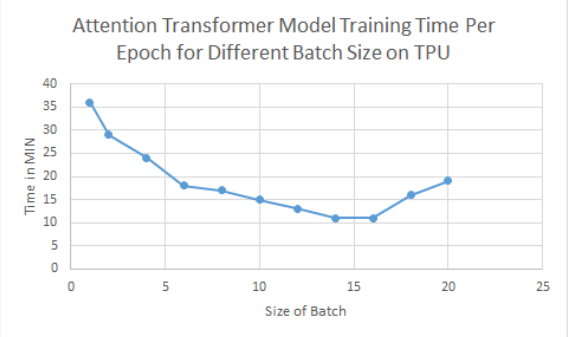


Figure 3: The figure shows a line graph illustrating the training time per epoch for different batch sizes.

In addition to that, to keep the training time of Attention Transformer practical, with a TPU of 180 TFlops and 64 GB High Bandwidth Memory (HBM). We have set the number of initial RNN based Encoder-Decoder layers to one and the number of the second Transformer Encoder-Decoder layer to two. Plus, the number of attention heads in the multi-head attention layer of the Transformer encoder-decoder layer is set to 8, and all other hyper-parameters for transformer model are kept as suggested in the paper (Vaswani et al., 2017).

3.4 *Training the Model*

In training, we have not augmented the original form of the given sentences in the data-set. In the pre-processing step, we have only removed punctuation from the sentences and fed the filtered data to the model, which sums up to 35,215 sentence pairs. We have used the selected hyper-parameters from previous subsections to train the model, which are obtained by optimizing dev data. In addition to that, to keep track of our models' performance, we have used the Sparse Categorical Cross Entropy function as our loss function for evaluating training predictions, which is an integer version of the Categorical Cross Entropy function the details can be observed in the notebook (implementation)

3.5 *Dealing with Over-Fitting and Unseen Vocabulary at Test Time*

To save the model from overfitting, we have kept the training procedure straight-forward by applying a simple rule to train the model until it provides a BLEU score of 0.7 or the performance of BLEU score asymptotes after ten epochs. While training, we have collected average BLEU scores, TER score, Accuracy, and Mean Loss across batches over an epoch to track the performance of the model as shown in (implementation).

Moreover, to deal with new input vocabulary at test time, we have employed a simple trick by generating a miscellaneous token at the time of tokenization. The miscellaneous token gets included in the vocabulary of the model at train time. And the model learns to deal with this token based on its neighbors. During test time, while tokenization, if the input sentence contains any unseen vocabulary, then we exchange that word with the miscellaneous token.

3.6 *Comparison With Baseline Model*

We have trained the baseline models and our proposed attention transformer model on the Marathi-Hindi data-set at the end of each epoch, we have recorded the training time. This per epoch training time will allow us to measure the quickness in the model to finish a training epoch. Fig. 4 below states the comparison of cumulative training time of all three models. It can be seen clearly that the Bahdanau model takes a huge amount of training time as compared to the other two models. We were able to run only 10 epochs for the Bahdanau model in approximately 9 hours. On the other hand Transformer and Attention, Transformer models are very quick, it takes approximately 7 minutes to train an epoch of both the models. However, over the time it can be seen in the graph that the baseline Transformer model is slightly quicker than our proposed Attention Transformer model.

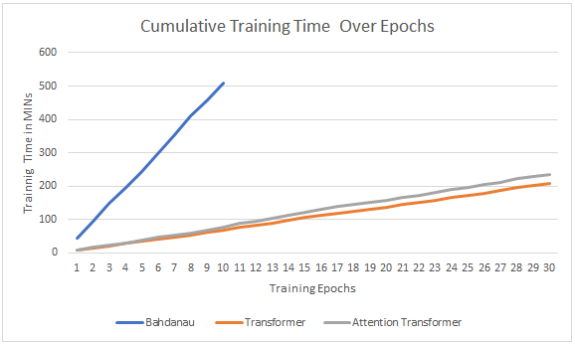


Figure 4: The figure shows comparison of cumulative training time of Bahdanau, Transformer and Attention Transformer model.

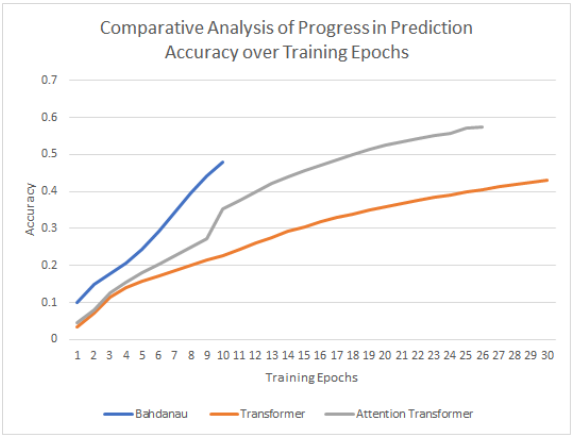


Figure 5: The figure shows comparison of progress in Sparse Categorical Accuracy as we continue to train the Bahdanau, Transformer and Attention Transformer model.

Next, to measure the progress in translation performance as we continue to train the model, we have recorded average Sparse Categorical Accuracy, BLEU score, Sparse Categorical cross entropy loss, and TER scores at the end of each epoch as shown in Fig. (5, 6, 7, 8) respectively. This track of per epoch training performance helps us to visualize the progress of the model in learning the translation probability distribution, plus we can also utilize this information to find out the most active model that fits translation distribution in the least number of epochs.

In the Fig. 5 and 6, we can notice that the Bah-

danau model is quickest to adapt translation probability distribution compared to other models as it has shown approximately exponential increment in accuracy and BLEU scores over the initial training epochs. The transformer model is following a relatively linear path in learning the probability distribution.

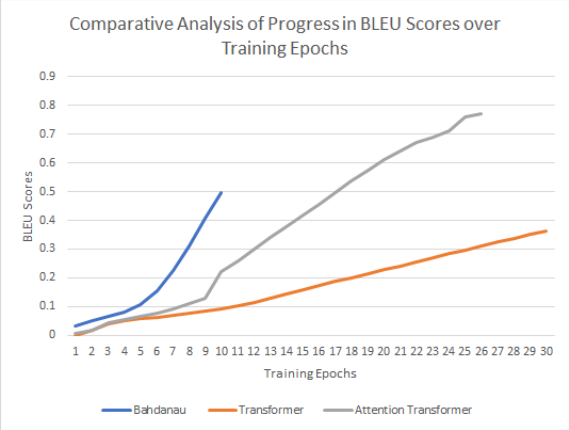


Figure 6: The figure shows comparison of progress in BLEU score as we continue to train the Bahdanau, Transformer and Attention Transformer model.

The reason behind the wining performance of the Bahdanau model is it's inherent recursive nature to model the sequence to sequence tasks, which helps it to learn the positional order of the given sequence. The Transformer lacks this recursive nature and uses a sinusoidal positional encoding scheme to get the awareness of the position of a word in a sentence, which is not as effective as Bahdanau's inherent recursive nature. But this recursive nature hinders the Bahdanau model to exploit parallelism due to this Bahdanau model takes more time to finish a training epoch as compare to the Transformer model.

The Attention Transformer takes the benefits of both the Bahdanau and the Transformer model. The initial layer of RNN helps the Attention Transformer to learn the positional order of the given sequence, plus the stacked Transformer above it allows the Attention Transformer to apply maximum parallelism. In the Fig. 5 and 6, we can notice that the Attention Transformer has given a relatively intermediary performance as compare to the other two models because we have kept the number of RNN and Transformer layers almost equal. If we increase the number of RNN layers in the Attention Transformer model it will start behaving more like the Bahdanau model likewise, if we increase the number of transformer layers then it will act more like the Transformer model.

Similarly, we can use the same argument to reason about the displayed behavior of the performance of the Bahdanau, Transformer, and Attention Transformer model in Fig. 7 and 8

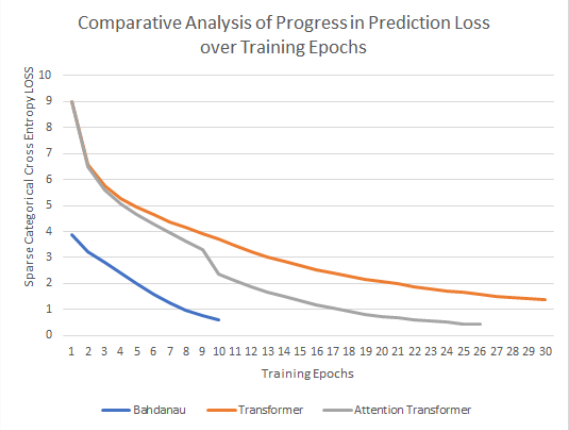


Figure 7: The figure shows comparison of decrements in loss as we continue to train the Bahdanau, Transformer and Attention Transformer model.

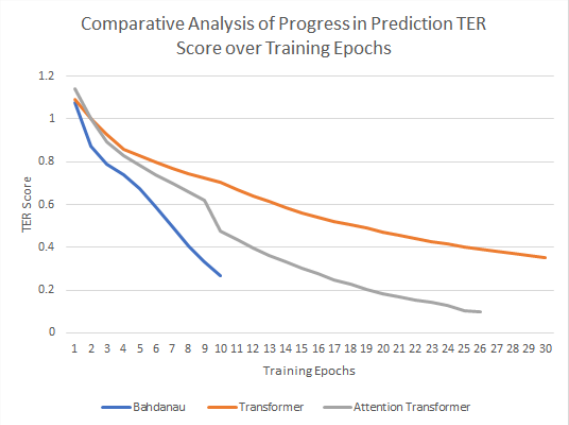


Figure 8: The figure shows comparison of decrements in TER score as we continue to train the Bahdanau, Transformer and Attention Transformer model.

Finally, we have utilized the trained Bahdanau, Transformer, and Attention Transformer model to translate 100 unseen records from Marathi to Hindi, which we have initially separated from the dev dataset. The Fig. 9 below displays the performance of the trained models on the scale of 0-1 BLEU points, the Bahdanau model fails to capture the distribution of unseen data, while the Transformer model performs relatively good. Attention Transformer gives comparatively better performance on average as it shows high BLEU scores for many of the instances.

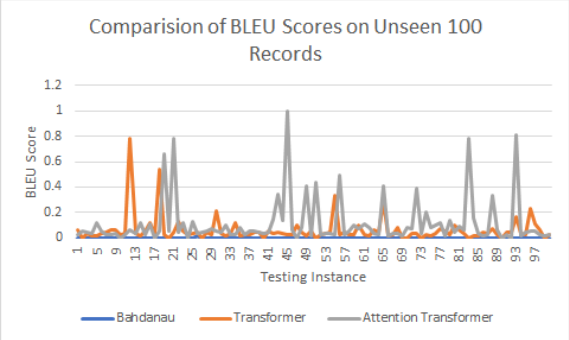


Figure 9: The figure shows comparison of calculated BLEU scores on the scale of 0-1 from the predictions of Bahdanau, Transformer and Attention Transformer model on the unseen 100 records which we have separated from the development data.

3.7 *Testing Results*

We have developed two instance models of Attention Transformer using the procedure mentioned above for both predicting Marathi sentences when Hindi sentences are given as input and predicting Hindi sentences when Marathi sentences are provided as input. We have achieved BLEU points of 3.68 and a TER score of 97.64 for the Hindi-Marathi test pair. Plus, BLEU points of 9.02 and the TER score of 88.68 for the Marathi-Hindi test data-set.

4 Conclusion

This paper has presented a supervised deep neural translation-based approach called Attention Transformer as a tool to perform translation between similar pair of languages (Hindi-Marathi). We have developed a novel Neural Translation method called Attention Transformer to transmute from Hindi source to Marathi and vice-versa by combining the classical recurrence based encoder-decoder approach and Transformers working mechanisms. All supervised translation approaches need parallel corpora as their data-set to learn the probability function of generating translation from source to target. We have solely utilized the WMT-20 Hindi-Marathi parallel corpus as the training data-set for the Attention Transformer model having 44,685 sentence pairs and used two human-level evaluation criteria, BLEU plus TER scores, to evaluate the Attention Transformer model. We have achieved BLEU points of 3.68 and a TER score of 97.64 for the Hindi-Marathi test pair. And, BLEU points of 9.02 with the TER score of 88.68 for the Marathi-Hindi test data-set. The future work under this domain includes applying stochastic optimizations like a genetic algorithm to find the best possible combinations of hyper-parameter to model the probability distribution of source to the target language. Furthermore, we can also stack a reinforcement learning paradigm on a developed supervised neural translation model to create a self-autonomous personalized environment for learning the probability function, which continuously gets updated by taking real-time feedback from the user.

References

Dzmitry Bahdanau and et al. 2014. Neural machine translation by jointly learning to align and translate. Cite arxiv:1409.0473Comment: Accepted at ICLR 2015 as oral presentation.

Kyunghyun Cho and et al. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1724–1734. ACL.

Anirudh Goyal and et al. 2016. Professor forcing: A new algorithm for training recurrent networks. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 4601–4609.

Zhiheng Huang, Peng Xu, and et al. 2020. TRANS-BLSTM: transformer with bidirectional LSTM for language understanding. *CoRR*, abs/2003.07000.

TEAM implementation. Attention transformer hindi-marathi machine translation.

Kishore Papineni and Roukos et al. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Matthew Snover and Bonnie Dorr et al. A study of translation error rate with targeted human annotation. In *In Proceedings of the Association for Machine Transaltion in the Americas (AMTA 2006*.

TensorFlow. a. Tensorflow implementation for bahdanau model online page, accessed: 14.08.2020.

TensorFlow. b. Tensorflow implementation for transformer model online page, accessed: 14.08.2020.

Ashish Vaswani, Shazeer, and et al. 2017. Attention is all you need. In I. Guyon and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Transformer-based Neural Machine Translation System for Hindi – Marathi: WMT20 Shared Task

Amit Kumar, Rupjyoti Baruah, Rajesh Kumar Mundotiya, Anil Kumar Singh
Department of Computer Science & Engineering
Indian Institute of Technology (B.H.U.)
Varanasi, India
{amitkumar.rs.cse17, rupjyotibaruah.rs.cse18,
rajeshkm.rs.cse16, aksingh.cse}@iitbhu.ac.in

Abstract

This paper reports the results for the Machine Translation (MT) system submitted by the NLPRL team for the Hindi – Marathi Similar Translation Task at WMT 2020. We apply the Transformer-based Neural Machine Translation (NMT) approach on both translation directions for this language pair. The trained model is evaluated on the corpus provided by shared task organizers, using BLEU, RIBES, and TER scores. There were a total of 23 systems submitted for Marathi to Hindi and 21 systems submitted for Hindi to Marathi in the shared task. Out of these, our submission ranked 6th and 9th, respectively.

1 Introduction

In the last decade and a half, neural machine translation (NMT) (Sutskever et al., 2014) has achieved great success in automatically translating human language text, outperforming statistical machine translation (SMT) (Koehn et al., 2003). Both the system require very large corpus sizes to train and evaluate the results. They, however, don't work very well for low resource data (He et al., 2016; Koehn and Knowles, 2017; Dowling et al., 2018). Translation from or to low resource languages is the major challenges faced by today's NMT systems.

Different methods have been proposed to overcome the data sparsity problem for low resource languages by researchers around the world. These include using monolingual data (Wu et al., 2019), fine-tuning (Miceli Barone et al., 2017) the high resource monolingual and parallel data on low resource data, back translation (Hoang et al., 2018), etc. They succeed up to some extent, but the success is limited, as the reported results show when compared to those for resource rich languages.

In this paper, we use the Transformer network-based NMT system (Vaswani et al., 2017) because it is among the state of the art models for machine translation. The work reported for this shared task is an extension of the work done by (Kumar and Singh, 2019) for similar languages task for 2019, which had also used a transformer based NMT system.

2 Similar Languages

Two languages are considered similar or closely related if they are close relatives in terms of the linguistic family of the linguistic family tree (or forest), or if the speakers of the two languages are in close contact over a long period of time. Contact over a long period leads to the exchange of cognates and loanwords between the speakers, sometimes even grammatical constructs.

Leveraging the close similarity of languages is one way to overcome the problem of data scarcity. Using similar features between such languages and improving translation is one of the directions for research for low resource machine translation.

For this submission, the motives behind conducting the shared task experiments are:

- To find out whether it is advantageous to use transformer-based NMT for similar languages.
- Whether using the SentencePiece [1] library without tokenization is beneficial for translation between similar languages or not.

3 Submitted System

We submitted two systems, namely, Marathi→Hindi and Hindi→Marathi. Both are the NMT systems trained on a Transformer (Vaswani et al., 2017) network. In this experiment, we did not tokenize data using any tokenizer. We directly applied SentencePiece library on the corpus. We found that directly applying

[1] https://github.com/google/sentencepiece

Proceedings of the 5th Conference on Machine Translation (WMT), pages 393–395

Parameters	Value
Encoder and decoder layers	5
Encoder embedding dimension	512
Decoder embedding dimension	512
Encoder attention heads	2
Decoder attention heads	2
Dropout	0.4
Attention dropout	0.2
Optimizer	Adam
Learning rate scheduler	inverse sqrt
Learning rate	1e-3
Minimum learning rate	1e-9
Adam-betas	(0.9, 0.98)
Number of epochs	100

Table 1: Hyperparameters used in our experiment

SentencePiece for preprocessing of data gives a better result. Since both the languages come under the category of morphologically rich and similar languages, directly applying SentencePiece on their corpus is advantageous. SentencePiece breaks the sentences into morphemes and phonemes. It extracts loanwords and cognate pairs. Breaking of sentences into subwords helps the neural translation network to learn better translations, and to generalize this knowledge to translate and produce unseen words, partly due to jointly developing the subword vocabulary.

4 Data

We trained the model on total 49434 number of Hindi - Marathi parallel corpus which belongs to three domains: News, PM India and Indic WordNet. Validation is done on total 1411 sentences. For testing, a total of 1941 sentences were used.

5 Experiment setup

We used fairseq [2] sequence to sequence encoder-decoder framework to train and evaluate the system. For hyper-parameter settings, we used the settings reported by (Guzmán et al., 2019) as these setting work well on low resource languages. Table 1 gives the hyper-parameter settings.

6 Results

Task organizers evaluate the systems using three evaluation metric: BLEU (Papineni et al., 2002), RIBES (Isozaki et al., 2010) and Translation Error Rate (TER) (Snover et al., 2006). We report the evaluation scores in table 2.

[2]https://github.com/pytorch/fairseq

system	BLEU	RIBES	TER
Marathi → Hindi	20.72	64.46	71.04
Hindi → Marathi	12.5	58.66	76.86

Table 2: Scores of our system evaluated by task organizers

7 Conclusion

In this paper, we perform experiments for translation between two similar languages: Hindi and Marathi. We submitted two systems: Marathi→Hindi and Hindi→Marathi, which were evaluated using BLEU, RIBES and TER. We found that SentencePiece works well for similar languages because it helps the Transformer in capturing the relations between two languages by providing morphemes, phonemes, cognate pairs, loanwords, etc. There were a total 23 systems submitted for Marathi → Hindi and 21 systems submitted for Hindi → Marathi in the shared task. Out of these, our system ranked 6th and 9th for Marathi → Hindi and Hindi → Marathi, respectively, considering the BLEU scores.

References

Meghan Dowling, Teresa Lynn, Alberto Poncelas, and Andy Way. 2018. Smt versus nmt: Preliminary comparisons for irish.

Francisco Guzmán, Peng-Jen Chen, Myle Ott, Juan Pino, Guillaume Lample, Philipp Koehn, Vishrav Chaudhary, and Marc'Aurelio Ranzato. 2019. The flores evaluation datasets for low-resource machine translation: Nepali-english and sinhala-english. *arXiv preprint arXiv:1902.01382*.

Wei He, Zhongjun He, Hua Wu, and Haifeng Wang. 2016. Improved neural machine translation with smt features. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, page 151–157. AAAI Press.

Vu Cong Duy Hoang, Philipp Koehn, Gholamreza Haffari, and Trevor Cohn. 2018. Iterative back-translation for neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 18–24, Melbourne, Australia. Association for Computational Linguistics.

Hideki Isozaki, Tsutomu Hirao, Kevin Duh, Katsuhito Sudoh, and Hajime Tsukada. 2010. Automatic evaluation of translation quality for distant language

pairs. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 944–952, Cambridge, MA. Association for Computational Linguistics.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39, Vancouver. Association for Computational Linguistics.

Philipp Koehn, Franz J. Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 127–133.

Amit Kumar and Anil Kumar Singh. 2019. NLPRL at WAT2019: Transformer-based Tamil – English indic task neural machine translation system. In *Proceedings of the 6th Workshop on Asian Translation*, pages 171–174, Hong Kong, China. Association for Computational Linguistics.

Antonio Valerio Miceli Barone, Barry Haddow, Ulrich Germann, and Rico Sennrich. 2017. Regularization techniques for fine-tuning in neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1489–1494, Copenhagen, Denmark. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of association for machine translation in the Americas*, volume 200. Cambridge, MA.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

Lijun Wu, Yiren Wang, Yingce Xia, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. 2019. Exploiting monolingual data at scale for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4207–4216, Hong Kong, China. Association for Computational Linguistics.

Hindi-Marathi Cross Lingual Model

Sahinur Rahman Laskar, Abdullah Faiz Ur Rahman Khilji, Partha Pakray, Sivaji Bandyopadhyay
Department of Computer Science and Engineering
National Institute of Technology Silchar
Assam, India
{sahinur_rs, abdullah_ug, partha}@cse.nits.ac.in, sivaji.cse.ju@gmail.com

Abstract

Machine translation (MT) is a vital tool for aiding communication between linguistically separate groups of people. The neural machine translation (NMT) based approaches have gained widespread acceptance because of its outstanding performance. We have participated in WMT20 shared task of similar language translation on Hindi-Marathi pair. The main challenge of this task is by utilization of monolingual data and similarity features of similar language pair to overcome the limitation of available parallel data. In this work, we have implemented NMT based model that simultaneously learns bilingual embedding from both the source and target language pairs. Our model has achieved Hindi to Marathi bilingual evaluation understudy (BLEU) score of 11.59, rank-based intuitive bilingual evaluation score (RIBES) score of 57.76 and translation edit rate (TER) score of 79.07 and Marathi to Hindi BLEU score of 15.44, RIBES score of 61.13 and TER score of 75.96.

1 Introduction

MT is a well-known task of natural language processing (NLP) wherein automatic translation is performed between different languages. Broadly, MT is categorized into rule-based and corpus-based, where rule-based is based on a pre-defined rules on the concerned languages and corpus-based finds a generalized approach after being trained on a large corpus. MT switches from rule-based approach to the corpus-based which blots out the need for linguistic expertise. In the corpus-based approach, example-based machine translation (EBMT), statistical machine translation (SMT) and NMT techniques are available. The disadvantage of EBMT is that even though the corpus is large, all examples are not covered. To mitigate the issues of the contemporary approach SMT is introduced Brown et al. (1990); Koehn (2010). The SMT based system makes an assumption based on probability scores of the translated text. And hence, the ranking is done. SMT also faces many issues like system complexity, long term dependency problem, context-analyzing inability, word-alignment and the rare word problem. The inefficiency of SMT leads to the development of the NMT Devlin et al. (2014). But like SMT, the NMT based model also suffers the requirement of sufficient training parallel corpus, which is a challenge in the case of low resource languages. For this reason, there is a demand for direct translation among similar language pairs by utilizing similarity features and monolingual data, so that less availability of the parallel data does not pose a challenge. However, the NMT technique achieves state-of-the-art approach in MT because of its transformer model Vaswani et al. (2017). For low resource language pair translation, NMT models have been improved with monolingual corpus Sennrich et al. (2016b); Burlot and Yvon (2018); Wu et al. (2019). In this work, we have adopted cross-lingual language model (XLM) Conneau and Lample (2019) to implement an NMT model for Hindi-Marathi similar language translation task because XLM shows significant improvements for low-resource languages by utilizing the monolingual corpora.

2 Related Work

Hindi-Marathi translation lacks background work. However, similar work is found on Hindi-Nepali pair at WMT19 shared task of similar language translation Laskar et al. (2019). The literature survey mainly focuses on NMT for low resource language pairs since NMT outperforms conventional SMT on low resource pairs like English to Mizo, English to Hindi, English to Punjabi, and English to Tamil Pathak et al. (2018); Pathak and Pakray (2018); Laskar et al. (2019). It is noticed that train-

Proceedings of the 5th Conference on Machine Translation (WMT), pages 396–401

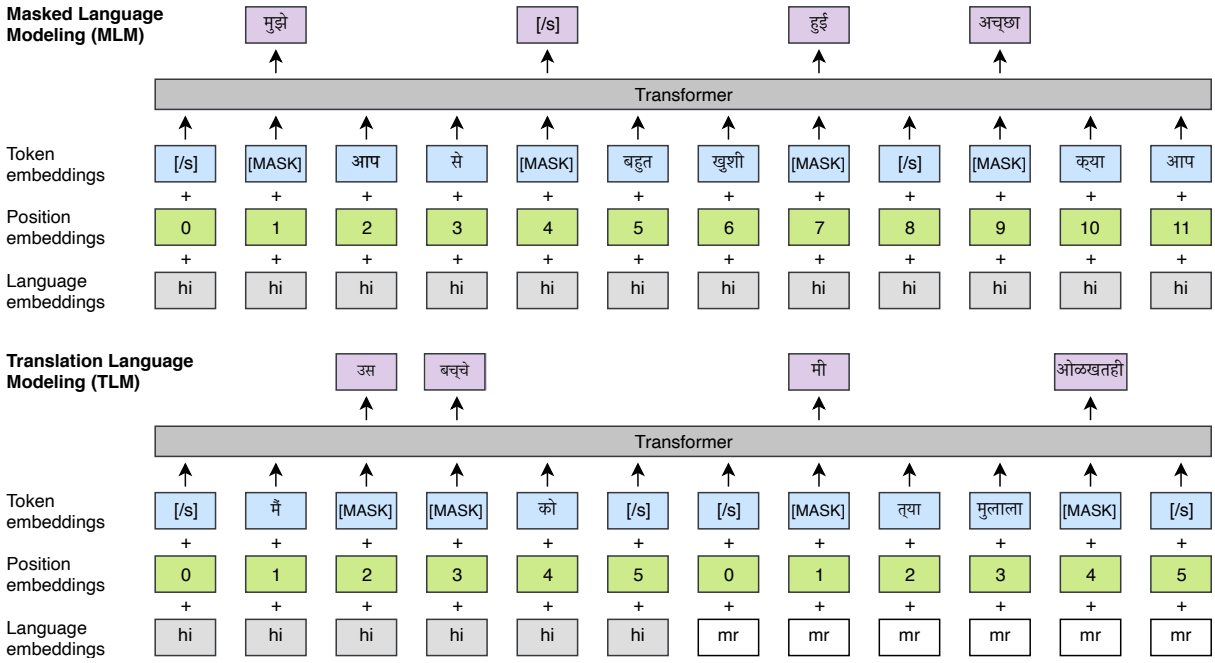


Figure 1: MLM pre-training inspired from Devlin et al. (2018) and TLM fine-tuning objective which extends the MLM task to parallel sentences as used by Conneau and Lample (2019). Diagram adapted from (Conneau and Lample, 2019) after suitable changes.

ing performance improves while parallel training data increases. For low resource languages, it is difficult to collect parallel data unlike monolingual data which is easily found through online sources. Hence, monolingual based NMT systems are introduced to enhance the translation quality of low resource language pair translation Sennrich et al. (2016b); Burlot and Yvon (2018); Wu et al. (2019). To get the advantage of monolingual data, unsupervised pre-train methods are introduced Ramachandran et al. (2017); Variš and Bojar (2019). Conneau and Lample (2019) proposed XLM based on bidirectional encoder representations from transformers (BERT) where the contextual language model is built with words based on preceding and succeeding context. No work has been done on Hindi-Marathi low resource language pair with such advanced NMT based approach, from the best of our knowledge. Our work investigates XLM model on Hindi-Marathi low resource language pair translation.

3 Dataset

3.1 Description

The organizers of WMT20 provided parallel and monolingual corpus for both Hindi and Marathi. The training dataset available for the WMT20, Hindi-Marathi task was obtained from three main sources viz. Indic WordNet, News, and PM India. Having 11,188, 12,349, and 25,897 parallel sentences (total 49434 sentences) respectively. The validation and test set contain 1941 and 1411 sentences. The Hindi monolingual dataset contains about 96 million sentences at about 32GB whereas the Marathi dataset is much smaller at only 4.72 million sentences totalling to around 2GB of corpus.

3.2 Preprocessing

We have removed unwanted symbols like URLs, email IDs and English text from the monolingual corpora of both the languages if any were to be present. In addition to this, since Hindi and Marathi languages share many common Devnagiri characters and hence to leverage this idea we have pre-processed the dataset obtained from Section 3.1 by a common vocabulary prepared via byte pair encoding (BPE) Sennrich et al. (2016a) on the same data provided by the organizer. Such an approach greatly helps in aligning the embedding space as shown in Lample et al. (2017). BPE learning is performed as used by Conneau and Lample (2019). The BPE is thus learnt after joining random sentences from the monolingual corpora. Following Conneau and Lample (2019) the text is sampled using a multinomial distribution. The distribu-

tion is as shown in Equation 1. The probabilities of the distribution are $p_{i=1...N}$. The BPE codes are generated and applied using the C++ implementation[1] of Sennrich et al. (2015).

$$p_i = \frac{q_i^\alpha}{\sum_{j=1}^{N} q_j^\alpha} \quad (1)$$

and p_i is as defined in Equation 2.

$$q_i = \frac{n_i}{\sum_{k=1}^{N} n_k} \quad (2)$$

α is taken as 0.5.

4 System Description

Our approach consists of the two principal approaches viz. the pre-training step and the fine-tuning step which are discussed in the following sub-sections 4.1 and 4.2. .

4.1 Pretraining our Model

For the pre-training step we have followed the steps of (Conneau and Lample, 2019) and utlized the masked language modeling (MLM) objective of (Devlin et al., 2018). Thus, following the work of Devlin et al. (2018) we have sampled 15% of BPE tokens randomly from the textual data and masked then by a `[MASK]` token roughly 80%. Also from the remaining 20%, the 10% component is randomly replaced and the rest part remains unchanged. The difference our approach has from the work of Devlin et al. (2018) is that, we have used lengths truncated to a fixed number (256 in our case), whereas the former uses pairs of sentences. To create a balance between the rare and commonly occuring BPE tokens like punctuation marks, the frequent outputs were subsampled using a multinomial distribution, where the weights are proportional to the inverse square root of the frequencies (an approach similar to Mikolov et al. (2013)). The pretraining objective is illustrated in Figure 1.

4.2 Fine Tuning

The model pre-training step follows an unsupervised approach and requires only the monolingual data. Since, the principal task for our work was to build a MT system, we need to leverage parallel data. Following, (Conneau and Lample, 2019) we used the translation language modeling (TLM) for fine-tuning the model obtained from Section 4.1. Here, instead of the truncated monolingual corpora we utlize the concatenation of parallel data as shown in Figure 1. Since the parallel sentences are concatenated for the concerned TLM task, we can mask and predict simultaneously from both Hindi and Marathi sentences. Enabling better placement of Hindi and Marathi word representations. Specifically as shown by Conneau and Lample (2019), this enables the model to leverage the context even if single handedly the source or target sentence is insufficient to decipher the sentence.

5 Experimental Setup

We have trained the transformer based cross language model (XLM) (Conneau and Lample, 2019) also known as MLM + TLM task. We have used 6 layers with 8 attention heads. An embedding layer is also used with size 256. Given the comparatively smaller Marathi dataset as discussed in Section 3.1, and limited availability of computational resources[2] we trained the smaller model instead of the usual 12 layers and 16 attention heads as proposed by Conneau and Lample (2019). Batch size of 32 was used. Following settings of Conneau and Lample (2019), attention dropout was set to 0.1, gelu activation was used. Also, adam was used as an optimizer with an initial learning rate of 0.0001. Rest of the parameters are same as used by Conneau and Lample (2019) in their experiments and as given in their GitHub repository[3].

6 Result and Analysis

The WMT20 organizer declared result for the shared task of similar language translation on Hindi to Marathi[4] and Marathi to Hindi[5] and the results of our system's is reported in Table 3. Our team's name is NITS-CNLP. The participated systems are evaluated by BLEU Papineni et al. (2002), RIBES Isozaki et al. (2010) and TER Snover et al. (2006) and the tracks are ranked by BLEU score. A total of 21 teams participated in Hindi to Marathi translation track and 23 teams for Marathi to Hindi translation track including both primary and contrastive system types. Our system's rank is 10 with BLEU score 11.59 for Hindi to Marathi translation

[1] https://github.com/glample/fastBPE

[2] The model was trained on a Quadro P200 GPU having 5GB of GPU RAM

[3] https://github.com/facebookresearch/XLM

[4] http://mzampieri.com/workshops/wmt/HI-MR.pdf

[5] http://mzampieri.com/workshops/wmt/MR-HI.pdf

Type	Source: Hindi Target: Marathi	
Short	Source Test Sentence	अवसरों की समानता है।
	Predicted Test Sentence	संधीची समानता आहे.
	Google Translation	संधींची समानता आहे.
	Bing Translation	संधींची समानता आहे.
Medium	Source Test Sentence	यह मेरे लिए एक बहुत ही सुखद अनुभूति रही है।
	Predicted Test Sentence	ही माझ्यासाठी अतिशय सुखद अनुभूती आहे.
	Google Translation	ही माझ्यासाठी खूप आनंददायी भावना आहे.
	Bing Translation	माझ्यासाठी ही खूप सुखद भावना आहे.
Long	Source Test Sentence	बल्कि यह एक सकारात्मक शांति है जहां हम सब करुणा और ज्ञान के आधार पर संवाद, सद्भाव और न्याय को बढ़ावा देने के लिए काम करते हैं।
	Predicted Test Sentence	ही एक सकारात्मक शांतता आहे जिथे आपण करुणा आणि ज्ञानाच्या आधारे संवाद सद्भावनेला प्रोत्साहन देण्यासाठी काम करतो.
	Google Translation	उलट ही एक सकारात्मक शांती आहे जिथे आपण सर्व करुणा आणि ज्ञानावर आधारित संवाद, सुसंवाद आणि न्यायाला चालना देण्यासाठी कार्य करीत आहोत.
	Bing Translation	उलट ही एक सकारात्मक शांती आहे जिथे आपण सर्वजण करुणा आणि ज्ञानावर आधारित संवाद, सामंजस्य आणि न्याय ाला प्रोत्साहन देण्याचे काम करतो.

Table 1: Best Performance examples for Hindi to Marathi translation.

Type	Source: Hindi Target: Marathi	
Long	Source Test Sentence	साथियो, जीएसटी की व्यवस्था को और सशक्त, और सरल करने के प्रयास लगातार चल रहे हैं।
	Predicted Test Sentence	मित्रांनो वस्तू आणि सेवा कर व्यवस्था अधिक सशक्त आणि सुलभ करण्याचे
	Google Translation	मित्रांनो, जीएसटी कारभारास आणखी बळकटी आणि सुलभ करण्यासाठी प्रयत्न सुरू आहेत.
	Bing Translation	मित्रांनो, जीएसटी प्रणाली अधिक सक्षम करण्यासाठी, सोपे करण्यासाठी प्रयत्न सुरू आहेत.

Table 2: Worst Performance examples for Hindi to Marathi translation.

Translation	System Type	BLEU	RIBES	TER
Hindi to Marathi	Primary	11.59	57.76	79.07
Marathi to Hindi	Primary	15.44	61.13	75.96

Table 3: Our system's results.

track and for Marathi to Hindi translation track, the rank is 15 with BLEU score 15.44 in primary configuration.

Analysis We have attained a lower BLEU score for Hindi to Marathi translation as compared to Marathi to Hindi translation as shown in Table 3. This is because we have used more Hindi monolingual corpus than Marathi monolingual corpus. As a result of this our NMT system encoded more frequency of Hindi words as compared to Marathi words and thus, decoder could be able to generate better target Hindi words than Marathi target words. To examine the best performance, we have considered sample source test sentences and corresponding predicted, Google[6], Bing[7] translated sentences for Hindi to Marathi translation in three different types of sentences such as short, medium and long sentences as shown in Table 1. Table 2 shows the worst performance of our NMT system in case of long type sentences. In Table 2, Google translation is better than our predicted test sentence and Bing translation.

7 Conclusion and Future Work

Our NMT system adopts cross lingual model for a similar language translation task of Hindi-Marathi pair in both forward and backward directions. The evaluated result and in-depth analysis of the predicted sentences shows that our NMT system performs well for the short and medium types of sentences and shows poor performance in long sentences. However, our NMT system needs more Marathi monolingual corpus and in the future works, multilingual NMT system will be developed to overcome the limitation of corpus for such low resource language pair translation.

Acknowledgement

Authors would like to thank WMT20 Shared task organizers for organizing this competition and also, thank Center for Natural Language Processing (CNLP) and Department of Computer Science and Engineering at National Institute of Technology, Silchar for providing the requisite support and infrastructure to execute this work.

[6]https://translate.google.co.in/

[7]https://www.bing.com/translator

References

Peter F. Brown, John Cocke, Stephen A. Della Pietra, Vincent J. Della Pietra, Fredrick Jelinek, John D. Lafferty, Robert L. Mercer, and Paul S. Roossin. 1990. A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85.

Franck Burlot and François Yvon. 2018. Using monolingual data in neural machine translation: a systematic study. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 144–155, Brussels, Belgium. Association for Computational Linguistics.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems*, pages 7059–7069.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Jacob Devlin, Rabih Zbib, Zhongqiang Huang, Thomas Lamar, Richard Schwartz, and John Makhoul. 2014. Fast and robust neural network joint models for statistical machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1370–1380, Baltimore, Maryland. Association for Computational Linguistics.

Hideki Isozaki, Tsutomu Hirao, Kevin Duh, Katsuhito Sudoh, and Hajime Tsukada. 2010. Automatic evaluation of translation quality for distant language pairs. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 944–952, Cambridge, MA. Association for Computational Linguistics.

Philipp Koehn. 2010. *Statistical Machine Translation*, 1st edition. Cambridge University Press, USA.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2017. Unsupervised machine translation using monolingual corpora only. *arXiv preprint arXiv:1711.00043*.

S. R. Laskar, A. Dutta, P. Pakray, and S. Bandyopadhyay. 2019. Neural machine translation: English to hindi. In *2019 IEEE Conference on Information and Communication Technology*, pages 1–6.

Sahinur Rahman Laskar, Partha Pakray, and Sivaji Bandyopadhyay. 2019. Neural machine translation: Hindi-Nepali. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 202–207, Florence, Italy. Association for Computational Linguistics.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318, Stroudsburg, PA, USA. Association for Computational Linguistics.

Amarnath Pathak and Partha Pakray. 2018. Neural machine translation for indian languages. *Journal of Intelligent Systems*, pages 1–13.

Amarnath Pathak, Partha Pakray, and Jereemi Bentham. 2018. English–mizo machine translation using neural and statistical approaches. *Neural Computing and Applications*, 30:1–17.

Prajit Ramachandran, Peter Liu, and Quoc Le. 2017. Unsupervised pretraining for sequence to sequence learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 383–391, Copenhagen, Denmark. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Edinburgh neural machine translation systems for wmt 16. *arXiv preprint arXiv:1606.02891*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *In Proceedings of Association for Machine Translation in the Americas*, pages 223–231.

Dušan Variš and Ondřej Bojar. 2019. Unsupervised pretraining for neural machine translation using elastic weight consolidation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 130–135, Florence, Italy. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Lijun Wu, Yiren Wang, Yingce Xia, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. 2019. Exploiting monolingual data at scale for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4207–4216, Hong Kong, China. Association for Computational Linguistics.

Transfer Learning for Related Languages: Submissions to the WMT20 Similar Language Translation Task

Lovish Madaan, Soumya Sharma, Parag Singla
Indian Institute of Technology, Delhi
{lovish97, soumyasharma98}@gmail.com, parags@cse.iitd.ac.in

Abstract

In this paper, we describe IIT Delhi's submissions to the WMT 2020 task on Similar Language Translation for four language directions: Hindi ↔ Marathi and Spanish ↔ Portuguese. We try out three different model settings for the translation task and select our primary and contrastive submissions on the basis of performance of these three models. For our best submissions, we fine-tune the mBART model (Liu et al., 2020) on the parallel data provided for the task. The pre-training is done using self-supervised objectives on a large amount of monolingual data for many languages. Overall, our models are ranked in the top four of all systems for the submitted language pairs, with first rank in Spanish → Portuguese.

1 Introduction

Machine Translation (MT) is currently tackled using rule-based methods (RBMT) (Charoenpornsawat et al., 2002), phrase-based statistical methods (SMT) (Koehn et al., 2003) and neural methods (NMT) (Cho et al., 2014; Sutskever et al., 2014; Bahdanau et al., 2015; Vaswani et al., 2017).

NMT has achieved high translation quality for several language pairs (Bojar et al., 2018; Barrault et al., 2019), but this level of performance usually requires large amounts of aligned data in the order of millions of sentence pairs. For low and medium resource languages, SMT performs better than NMT (Koehn and Knowles, 2017; Sennrich and Zhang, 2019). SMT also shows better performance when there is a domain mismatch between the train and test datasets, which is typical of low and medium resource language pairs.

In these settings, NMT performance can be boosted by leveraging additional monolingual data to enforce various types of constraints or increasing the training data using back-translation. These methods can be particularly helpful if the source and target languages in MT are closely related and share language structure and alphabet. Recently, pre-training methods for sequence-to-sequence (seq2seq) models have been introduced like MASS (Song et al., 2019a), XLM (Conneau and Lample, 2019), BART (Lewis et al., 2019), and mBART (Liu et al., 2020). These methods show significant gains in downstream tasks like NMT, summarization, natural language inference (NLI), etc. In this paper, we focus on the transfer learning capabilities in NMT for the task of translation between related languages where parallel data is scarce.

IIT Delhi participated in the WMT 2020 Shared task on Similar Language Translation for four language directions: `Hindi (hi)` ↔ `Marathi (mr)` and `Spanish (es)` ↔ `Portuguese (pt)`. The first language pair is low resource and second is medium resource in terms of the parallel data available for the task. Refer to Table 2 for the classification.

We fine-tuned the pre-trained mBART model (Liu et al., 2020) on the parallel data provided for the task. mBART gives better performance than SMT models even when the parallel data is very limited. mBART is pre-trained on 25 languages, which contain Hindi and Spanish, but not Marathi and Portuguese. mBART is able to leverage transfer learning capabilities even for those languages that are originally not present during the pre-training phase. The fine-tuned mBART architecture forms our best submissions for both language pairs: `hi` ↔ `mr` and `es` ↔ `pt`. The rankings obtained by us in each of the language directions are listed in Table 1. Our findings are in line with earlier observations in the literature where transfer learning techniques have been shown to significantly boost NMT performance.

The rest of the paper is organized as follows: Section 2 provides the background and related work for low/medium resource NMT. Section 3 gives an

Proceedings of the 5th Conference on Machine Translation (WMT), pages 402–408
Online, November 19–20, 2020.

Direction	BLEU	Rank
hi → mr	15.14	4
mr → hi	24.53	2
es → pt	32.69	1
pt → es	32.84	2

Table 1: BLEU scores on the test set provided for the task and system rankings according to the automatic evaluation metrics.

overview of the systems tried. In Section 4, we present the experiments and training pipeline setup. The results and analysis are detailed in Section 5. We finally conclude in Section 6.

2 Background

SMT is tackled by building a phrase table from the aligned parallel data. The target side translation is then generated by matching the most appropriate phrases in the source sentence conditioned on the target side language model along with a reordering model (Koehn et al., 2003).

NMT is modeled using Encoder-Decoder models (Cho et al., 2014; Sutskever et al., 2014; Bahdanau et al., 2015), with the Transformer model (Vaswani et al., 2017) achieving state-of-the-art on many MT problems. But these models' reliance on large aligned parallel data for the source and target languages makes them unsuitable for low/medium resource language pairs (Koehn and Knowles, 2017). Some of the previous works in these settings to improve NMT performance are described below:

2.1 Multilingual NMT

Instead of using only two languages (source and target) for training an NMT model, using multiple languages has been shown to help in low resource scenarios. For example, it might be the case that a certain pair of languages have very little parallel data between them, but there exists a third language with abundant parallel data with the original two languages. This third language acts as a pivot and helps in improving NMT between the two languages (Aharoni et al., 2019; Gu et al., 2018; Liu et al., 2020; Zhang et al., 2020).

2.2 Back-Translation

Back-Translation (Sennrich et al., 2016; Edunov et al., 2018; Hoang et al., 2018) increases the amount of training data by using monolingual corpus along with partially-trained NMT models on the limited parallel data. Pseudo-parallel corpus for each direction is first obtained by generating the translations of the monolingual data for each language using the partially-trained MT models on the limited parallel data. Using these pseudo-parallel corpora, the partially-trained NMT models are then trained further for some number of steps. In this way, millions of pseudo-parallel sentence pairs can be generated to improve NMT models because of the abundance of monolingual data. Another version of using back-translation is the copying mechanism. Currey et al. (2017) proposes to copy the target side monolingual data on the source side to create additional data without modifying the training regimen for NMT. This helps the model to generate fluent translations.

2.3 Pre-trained Language Models

For NMT, the first step is the random initialization of model weights in both the encoder and decoder. Instead of random initialization, NMT models can be initialized by pre-training parts of the model (Conneau and Lample, 2019; Edunov et al., 2019), or pre-training the complete seq2seq model (Ramachandran et al., 2017; Song et al., 2019b; Liu et al., 2020). These pre-training methods leverage different kinds of masking techniques and the pre-training objective is to predict these masked tokens, similar to BERT (Devlin et al., 2019). Denoising auto-encoding can also be used where a sentence is corrupted by various noising techniques and the pre-training objective is to generate the original uncorrupted sentence as in BART (Lewis et al., 2019) and mBART (Liu et al., 2020).

2.4 Incorporating Linguistic Information in NMT

There also have been works to improve low/medium resource NMT by adding linguistic information either using data augmentation (Currey and Heafield, 2019), subword embedding augmentation (Sennrich and Haddow, 2016) , or architectural changes (Eriguchi et al., 2017). This helps the model to not only learn the alignment between source and target language spaces, but also syntax structure like dependency parse, part of speech, etc. This helps in making the target side translations more fluent and conforming to the structure of the language. We do not explore this direction in this paper.

3 System Overview

We experimented with three different settings for hi ↔ mr as listed below.

SMT This phrase-based system leverages both monolingual and parallel data provided for the task. We use Moses (Koehn et al., 2007) for training the SMT systems.

NMT (Transformer) For this, we used the standard Transformer large architecture from Vaswani et al. (2017) for training on the parallel data provided for the task.

NMT (mBART) mBART (Liu et al., 2020) is a large Transformer pre-trained on monolingual data for 25 languages. The pre-training objective for mBART is seq2seq de-noising for natural text as in BART (Lewis et al., 2019). mBART provides a general-purpose pre-trained Transformer for any downstream task. It has been shown to give significant improvements over the random initialization for NMT and is the current state-of-the-art for many low resource language pairs.

Implementation Details mBART uses a shared subword vocabulary of 250K tokens for all the 25 languages present in the pre-training. We use the same vocabulary for Marathi and Portuguese also, even though they were not used during the pre-training phase. Marathi shares its subword vocabulary with languages like Hindi and Nepali in mBART, and Portuguese shares with Spanish, Italian and other European languages present in mBART. The percentage of unknown tokens [UNK] in Marathi and Portuguese parallel datasets is less than 0.003% when using the shared mBART vocabulary.

Additionally, the mBART architecture requires language specific token at the end of each input sequence to provide the language specific context for the decoder. Since Marathi and Portuguese were not present during the pre-training phase, we use the token corresponding to the second most related language present in mBART pre-training for specifying the context at the time of decoding in each case. For Marathi, we used the Nepali language token and for Portuguese, we used the Italian language token. We could not use Spanish language token for Portuguese because we are doing translations to and from Spanish.

	train	valid	test
hi ↔ mr	43,274	1,411	1,941
es ↔ pt	3,472,860	1,283	1,495

Table 2: Dataset statistics. First is low resource pair (# train < 1 Million) and second is medium resource (1 Million < # train < 10 Million).

Model	**hi - mr**	
	←	→
SMT	18.74	14.91
mBART	**24.53**	**15.14**

Table 3: BLEU scores on Hindi ↔ Marathi on the test set for our primary and contrastive submissions.

4 Experiments

We use hi ↔ mr and es ↔ pt language pairs for our experiments.

4.1 Datasets & Preprocessing

Because of the constrained nature of the shared task, we only use the parallel data provided for this task. We removed the empty instances for both language pairs (< 2000 instances). For es ↔ pt, we do not use 'WikiTitles v2' part of the parallel data for training because of very short sentences in the dataset. The cleaned parallel dataset statistics are provided in Table 2.

Preprocessing We use sentence piece tokenization (Kudo and Richardson, 2018) for generating the source and target sequences for the NMT architectures. For the standard Transformer, we train a sentence piece model using 40K subword tokens for hi ↔ mr. For mBART, we use Liu et al. (2020)'s pre-trained[1] sentence piece model comprising of 250K subword tokens as the vocabulary.

For the SMT model on hi ↔ mr, we also use the monolingual data provided for this task. We extract 5 Million monolingual sentences each for Hindi and Marathi after deduplication and use this set for training the language models. We use Moses (Koehn et al., 2007) for all tokenization / detokenization scripts.

[1] https://github.com/pytorch/fairseq/blob/master/examples/mbart/README.md

Submission	hi - mr		es - pt	
	←	→	←	→
IIT Delhi (ours)	**24.53**	15.14	32.84	**32.69**
Rank 1	24.53	18.26	33.82	32.69

Table 4: Hindi - Marathi and Spanish - Portuguese BLEU scores on the test dataset of the Similar Language Translation Task. Our submission scores are bolded when they match the first ranked submission.

4.2 Model Architectures & Training

SMT We generate a phrase table for the SMT model using the code provided by Lample et al. (2018). We used Moses (Koehn et al., 2007) and Giza++ with standard settings to train the SMT model in both directions.

NMT (Transformer) We use the large Transformer from Vaswani et al. (2017) with 8 encoder and decoder layers and replicate all the parameters from Ott et al. (2018). The number of parameters in the model are approximately 248 Million and it takes ∼26 hours on 4 Nvidia V100 (32 GB) GPUs.

NMT (mBART) For this, we use 12 Transformer encoder and decoder layers, with total number of model parameters ∼611 Million. We use the pre-trained mBART for initializing the model weights. We follow the recommendations of Liu et al. (2020) for the hyperparameter settings. We stop the training after 25K gradient updates for the model. These updates take ∼35 hours on 4 Nvidia V100 (32 GB) GPUs.

4.3 Evaluation

We use case-insensitive BLEU scores (Papineni et al., 2002) calculated using sacreBLEU[2] (Post, 2018). These scores are calculated on the validation set to decide our primary and contrastive submissions. For evaluating performance on the test set, the organizers use BLEU, TER (Snover et al., 2006), and RIBES (Isozaki et al., 2010).

5 Results and Analysis

Results Table 3 shows our results on the test set for our primary and contrastive submissions. We observed the performance of our three model settings on the validation set, and we selected the mBART model as our primary submission and SMT model as the contrastive submission for `hi` ↔ `mr`. Similarly, the mBART model forms our primary submission for `es` ↔ `pt`. Table 4 lists our final results on this shared task. We also list the BLEU scores for the submission that got first rank in each of the language directions. Since the test sets were hidden at the time of submission, we do not report our numbers on the standard Transformer architecture.

Analysis Even though Marathi and Portuguese are not present during the pre-training phase of mBART, fine-tuning on these languages provides significant boosts over SMT and standard Transformer. This shows that some level of language independent multilingual embeddings are present in the pre-trained model weights which can be exploited for the transfer task.

6 Discussion and Conclusion

We have participated in the Similar Language Translation task on four language directions. We have shown that pre-trained models can help in low and medium resource NMT. Our best system uses the pre-trained mBART model (Liu et al., 2020) and fine-tunes on the parallel data provided for the specific translation task. Our results demonstrate that pre-training can help even when the language used for fine-tuning is not present during pre-training.

One direction of future work is to add linguistic information during the pre-training phase to get more fluent translations. When this information is not available directly (especially for low resource languages), pre-training on a related high resource language with syntax information can help low resource languages also.

Acknowledgments

We thank the IIT Delhi HPC facility[3] for the computational resources. We are also thankful to Ganesh Ramakrishnan and Pawan Goyal for initial discussions on the project. Parag Singla is supported

[2]Signature: BLEU + case.mixed + numrefs.1 + smooth.exp + tok.13a + version.1.3.1

[3]http://supercomputing.iitd.ac.in/

by the DARPA Explainable Artificial Intelligence (XAI) Program with number N66001-17-2-4032, Visvesvaraya Young Faculty Fellowships by Govt. of India and IBM SUR awards. Any opinions, findings, conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views or official policies, either expressed or implied, of the funding agencies.

References

Roee Aharoni, Melvin Johnson, and Orhan Firat. 2019. Massively multilingual neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 3874–3884. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Loïc Barrault, Ondrej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. 2019. Findings of the 2019 conference on machine translation (WMT19). In *Proceedings of the Fourth Conference on Machine Translation, WMT 2019, Florence, Italy, August 1-2, 2019 - Volume 2: Shared Task Papers, Day 1*, pages 1–61. Association for Computational Linguistics.

Ondrej Bojar, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Philipp Koehn, and Christof Monz. 2018. Findings of the 2018 conference on machine translation (WMT18). In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers, WMT 2018, Belgium, Brussels, October 31 - November 1, 2018*, pages 272–303. Association for Computational Linguistics.

Paisarn Charoenpornsawat, Virach Sornlertlamvanich, and Thatsanee Charoenporn. 2002. Improving translation quality of rule-based machine translation. In *COLING-02: Machine Translation in Asia*.

Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. In *Proceedings of SSST@EMNLP 2014, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation, Doha, Qatar, 25 October 2014*, pages 103–111. Association for Computational Linguistics.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 7057–7067.

Anna Currey, Antonio Valerio Miceli Barone, and Kenneth Heafield. 2017. Copied monolingual data improves low-resource neural machine translation. In *Proceedings of the Second Conference on Machine Translation, WMT 2017, Copenhagen, Denmark, September 7-8, 2017*, pages 148–156. Association for Computational Linguistics.

Anna Currey and Kenneth Heafield. 2019. Incorporating source syntax into transformer-based neural machine translation. In *Proceedings of the Fourth Conference on Machine Translation, WMT 2019, Florence, Italy, August 1-2, 2019 - Volume 1: Research Papers*, pages 24–33. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Sergey Edunov, Alexei Baevski, and Michael Auli. 2019. Pre-trained language model representations for language generation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4052–4059. Association for Computational Linguistics.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 489–500. Association for Computational Linguistics.

Akiko Eriguchi, Yoshimasa Tsuruoka, and Kyunghyun Cho. 2017. Learning to parse and translate improves neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 2: Short Papers*, pages 72–78. Association for Computational Linguistics.

Jiatao Gu, Hany Hassan, Jacob Devlin, and Victor O. K. Li. 2018. Universal neural machine translation for extremely low resource languages. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics:*

Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers), pages 344–354. Association for Computational Linguistics.

Cong Duy Vu Hoang, Philipp Koehn, Gholamreza Haffari, and Trevor Cohn. 2018. Iterative back-translation for neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation, NMT@ACL 2018, Melbourne, Australia, July 20, 2018*, pages 18–24. Association for Computational Linguistics.

Hideki Isozaki, Tsutomu Hirao, Kevin Duh, Katsuhito Sudoh, and Hajime Tsukada. 2010. Automatic evaluation of translation quality for distant language pairs. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, EMNLP 2010, 9-11 October 2010, MIT Stata Center, Massachusetts, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 944–952. ACL.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the association for computational linguistics companion volume proceedings of the demo and poster sessions*, pages 177–180.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation, NMT@ACL 2017, Vancouver, Canada, August 4, 2017*, pages 28–39. Association for Computational Linguistics.

Philipp Koehn, Franz J. Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 127–133.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018: System Demonstrations, Brussels, Belgium, October 31 - November 4, 2018*, pages 66–71. Association for Computational Linguistics.

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Phrase-based & neural unsupervised machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2019. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *CoRR*, abs/1910.13461.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *CoRR*, abs/2001.08210.

Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. 2018. Scaling neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 1–9, Brussels, Belgium. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA*, pages 311–318. ACL.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers, WMT 2018, Belgium, Brussels, October 31 - November 1, 2018*, pages 186–191. Association for Computational Linguistics.

Prajit Ramachandran, Peter J. Liu, and Quoc V. Le. 2017. Unsupervised pretraining for sequence to sequence learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 383–391. Association for Computational Linguistics.

Rico Sennrich and Barry Haddow. 2016. Linguistic input features improve neural machine translation. In *Proceedings of the First Conference on Machine Translation: Volume 1, Research Papers*, pages 83–91, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Rico Sennrich and Biao Zhang. 2019. Revisiting low-resource neural machine translation: A case study. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 211–221. Association for Computational Linguistics.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation.

In *Proceedings of Association for Machine Translation in the Americas, 2006*.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019a. MASS: masked sequence to sequence pre-training for language generation. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 5926–5936. PMLR.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019b. MASS: masked sequence to sequence pre-training for language generation. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 5926–5936. PMLR.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3104–3112.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 5998–6008.

Biao Zhang, Philip Williams, Ivan Titov, and Rico Sennrich. 2020. Improving massively multilingual neural machine translation and zero-shot translation. *CoRR*, abs/2004.11867.

The IPN-CIC team system submission for the WMT 2020 similar language task

Luis A. Menéndez-Salazar[1], **Grigori Sidorov**[1], **Marta R. Costa-Jussà**[2]
[1]Centro de Investigacion en Computacion
Instituto Politecnico Nacional, Mexico
[2]TALP Research Center
Universitat Politecnica de Catalunya, Barcelona
menendez.sla@gmail.com, sidorov@cic.ipn.mx, marta.ruiz@upc.edu

Abstract

This paper describes the participation of the NLP research team of the IPN Computer Research center in the WMT 2020 Similar Language Translation Task. We have submitted systems for the Spanish-Portuguese language pair (in both directions). The three submitted systems are based on the Transformer architecture and used fine tuning for domain Adaptation.

1 Introduction

In this paper we describe the Neural Machine Translation(NMT) systems developed by the NLP team of the Computer Research System of the Instituto Politécnico Nacional, México for the similar languages translation shared task of the EMNLP 2020 fifth conference on machine translation (WMT 20).

For this task, we submit systems for both directions of the Spanish-Portuguese language pair, all the submitted systems were based in a transformer architecture with a fine tuning for domain adaptation, the difference of the submitted systems was mainly the kind of tokens used (words and sub-word units) and the initialization of the word embeddings in the systems using either a random initialization or pre-trained word embeddings.

In the past year task, the submissions of the MLLP-UPV team (Baquero-Arnal et al., 2019) showed that the use of fine tuning was proven to be useful in this specific task, although the test corpus of this year had a higher complexity, the use of fine tuning for context adaptation was also beneficial for translation quality.

The paper was organized in the following way. Section 2 describes the architecture used for the systems trained, the section 3 describes the corpora used and the pre-processing of the texts, the section 4 gives a description of the training of the system, section 5 gives a description of the method used for obtaining the translations from the trained system, section 6 shows the results for both the internal evaluation and the evaluation of the task, section 7 is a discussion about the impact of pre-trained word embeddings in the submitted systems and section 8 presents the conclusions.

1.1 Transformer models in NMT

Before transformers most state-of-the-art MT systems relied on recurrent neural networks, with attention mechanisms, but the RNN based architectures although that in theory the information of each token can propagate arbitrary far down in the sequence, due to vanishing gradient in practice when dealing with long sentences or sequences information about the initial tokens can be lost.

As a solution of that problem transformers (Vaswani et al., 2017) are an architecture based in an encoder -decoder approach, but rely mainly in self attention mechanisms.

1.1.1 Encoder

Each encoder layer consists of two components: a self-attention mechanism and a feed-forward neural network. The self-attention mechanism receives a set of encoded representation from the previous layer and weights it in order to generate a set of output encodings. The feed forward network processes each output encoding and passes it to the next encoder and to the decoders.

The first encoder layer uses as arguments positional information and the word embeddings, instead of encodings.

1.1.2 Decoder

Each decoder layer has three main components: A self-attention mechanism, an over the encodings attention mechanism and a feed-forward network. The decoder layer works in a similar way than a encoder one, but the additional attention mecha-

Proceedings of the 5th Conference on Machine Translation (WMT), pages 409–413

nism uses the relevant information produced by the encoder layers.

In a similar structure from the first encoder layer, the first decoder layer also receives as inputs positional information and the embeddings of the output sequence. Due to the transformer should not know current or future information in order to predict the next word, the output sequence should be partially hidden during the training of the system.

The last decoder layer is followed by a linear transformation and a softmax layer to produce the probability of the words in the vocabulary.

1.2 Word embeddings initialization

Word embeddings are a solution in Natural Language Processing for the problem of having spare word spaces of high dimensionality that happens with the one-hot vectors representation.

Word embeddings uses a machine learning algorithm in order to learn the relations between words and contexts from big corpora, proposed from (Mikolov et al., 2013)

Inside of neural networks architecture approaches, word embeddings are generally used as a word representation in both source and target languages which are usually random initialized, but it is also possible to use pre-trained word embeddings and updated it in the training time.

Fast text (Bojanowski et al., 2016) is a library used to learn word embeddings, this model aims to create supervised and non-supervised systems to obtain word representation. It is also provided by the project pre-trained embeddings for 294 languages.

In the current paper an approach using fast-text vectors for the initialization of a transformer model is attempted, using the pre trained vectors in both Spanish and Portuguese languages.

2 Architecture of the submission

The main model for the submission consisted in a transformer model that used tokens composed by words and the word embeddings inside the transformer architecture were initialized using pre trained fast text embeddings in both, source and target languages.

For contrastive purposes two additional models were added to the submission , neither of them used a special initialization of the word embeddings and the main distinction between them was the kind of tokens used in the training, the first one used words and the later used sub word units gated by a BPE algorithm.

Corpus	Version	Sentences
JCR	1	1,650,126
Europarl	10	1,801,845
News commentary	15	48,259
Wikititles	2	649,833

Table 1: Training corpora

2.1 Model description

The three models for the submission differs in the following way:

1. **Primary:** Model that was initialized using pre trained fast-text word embeddings, and tokens constituted by words

2. **Contrastive1:** Model that was initialized with random word embeddings. Used tokens formed by words

3. **Contrastive2:** Model that was initialized with random word embeddings. Used tokens formed by BPE sub-word units

Where the primary model was the main model for the submission and the contrastive models serves as baselines.

For the comparative between the three models, BLEU (Papineni et al., 2002) was computed with the `Sacrebleu` (Post, 2018).

2.2 Transformer model

For the transformer model the configuration used consists of a model size of 6 layers, 512 feed-forward size, 8 heads, trained on one GPU with a batch size of 4096 tokens using. We stored a checkpoint every 5000 steps until 200000.

We used Adam optimizer with a β 2 of 0.998 The models were built using Open NMT toolkit (Klein et al., 2017).

3 Corpus description

The training data was made up with the available training data for the task, that is JCR, Europarl, news commentary and wikititles corpora. The provided development set was randomly split in two disjoint sets of the same size, dev1 and dev2 sets.

The data was prepossessed using the following pipeline tokenization, lowercasing and a BPE algorithm learned over the test set.

Model	Before	After
Primary	24.20	32.56
Contrastive1	23.94	30.43
Contrastive2	24.49	30.2

Table 2: BLEU for the models before and after fine-tuning for ES-PT

Model	Before	After
Primary	27.61	34.41
Contrastive1	27.21	34.11
Contrastive2	27.26	34.18

Table 3: BLEU for the models before and after fine-tuning for PT-ES

4 Training

The training for all the systems was carried in a two steps way.

In the first step the model was trained using the training set during 200 thousand steps, storing a checkpoint every 5000 steps. For all the checkpoint generated, a translation of the dev-set was computed and BLEU evaluated against a tokenized and lowercased version of the dev1 set.

Due to the conformation of the training data, that is made mostly of parlament sesions (Europarl) and scientific journals (JCR) and the observation that this domains doesn't appear in the test data there is an assumption of a domain mismatch between training and test data. Due to this mismatch in the second step a fine tunning of the model is conducted.

This fine tuning was trained from the best BLEU scored checkpoint and a retraining was made using the dev1 set up to 3000 steps, storing a checkpoint every 10. For all the stored checkpoints, a translation of the test set was computed and evaluated with BLEU against a tokenized and lowercased version of the dev2 set.

For the generation of the translations for the submission the checkpoint with the best BLEU score in the fine tuning step was used.

5 Translations generation

In order to get the translation for the evaluation the `translate.py` script of Open NMT was used with a beam of size 5 and a length penalization with alpha of 5.

After getting the translations the texts was passed through a recaser trained over the training corpora using the script `recaser.perl` from Moses, after this step a detokenizer was used.

6 Results

The tables 2 and 3 shows the BLEU obtained in the evaluation of the three different models before and after the fine tuning for the ES-PT and PT-ES language pairs respectively.

Model	BLEU	RIBES	TER
Primary	27.08	72.98	55.34
Contrastive1	23.91	71.55	57.55
Contrastive2	23.9	73.73	58.07

Table 4: Official results for submitted ES-PT systems

In this internal evaluation of the models the primary model outperforms the baseline models by 2.7 BLEU points for ES-PT direction and 0.18 points in PT-ES.

6.1 Task results

The evaluation of the task was carried using BLEU, RIBES (Isozaki et al., 2010) and TER (Snover et al., 2006) metrics the main difference between this measure and the internal one was that the internal evaluation used a tokenized lowercased version of the text and the task results used the final version.

The tables 4 and 5 show the results of the submitted systems in the task evaluation for the ES-PT and PT-ES language pairs respectively.

In this evaluation again the primary model outperforms the baseline models by a margin of 3 and 0.4 BLEU points for the ES-PT and PT-ES directions respectively.

7 Impact of the pre trained word embeddings

The pre-trained word embeddings used for the model were filtered in order to include only the words that was present in either the development set or the test set and preprocessed using the script `embeddings_to_torch.py` included in the Open NMT toolset.

The used word embeddings showed an im-

Model	BLEU	RIBES	TER
Primary	28.38	72.24	56.27
Contrastive1	27.98	72.11	56.16
Contrastive2	27.41	75.18	57.28

Table 5: Official results for submitted PT-ES systems

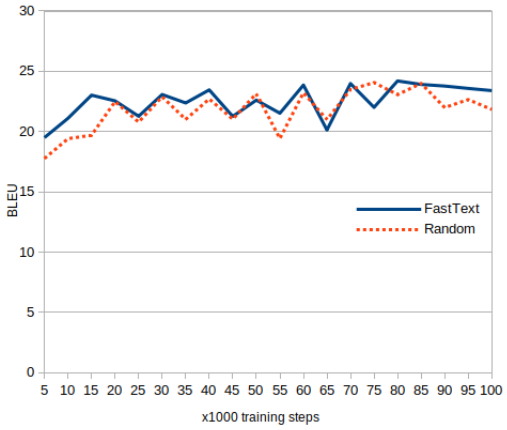


Figure 1: Training for ES - PT

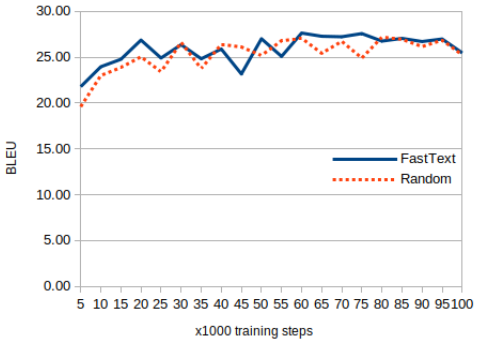


Figure 2: Training for PT - ES

provement in the translation quality (according to BLEU).

Figures 1 and 2 show comparatives for the ES-PT and PT-ES directions respectively for the first 100k training steps for the model 1 and 3, that means comparing both systems trained in words with the unique difference is that system 1 has fast text pre-trained word embeddings.

From the comparative in figures 1 and 2 we can extract that the use of pre-trained word embbedings was beneficial in the beginning of the training, with a difference of around 2 BLEU points in the first 5000 training steps for both ES-PT and PT-ES directions.

Similar results were seen in the fine tuning of the systems, in this case due to the short amounts of epoch for the training of this step a more detailed table is not provided, but in the tables 2 and 3 a difference of 2.7 and 0.3 BLEU points can be seen for the ES-PT and PT-ES directions respectively.

8 Conclusions

The initialization using fast text had a beneficial result in this low resource scenario, but the embeddings used were trained in a general context, is possible that pre-trained the embeddings in the specific context could gather better results.

In this specific experiment both contrastive models had similar results, independently of the kind of tokens used during the training.

Compared with the 2019 edition of the SLT task, this year the test corpus had a different domain, resulting in a lower BLEU score using similar techniques, but also in this year the use of fine tuning improved the translation margin in around 9 points for ES-PT and in almost 7 points for the evaluation using the development set in the primary models.

For the next year submission, the use of word embeddings can be expanded using word embeddings trained in a bilingual context or in a similar domain from the one in the test corpora.

9 Credits

The work was done with support of the Government of Mexico via CONACYT, SNI, CONACYT grant A1-S-47854, and grant SIP 20200797, of the Secretaría de Investigación y Posgrado of the Instituto Politécnico Nacional, Mexico.

References

Pau Baquero-Arnal, Javier Iranzo-Sánchez, Jorge Civera, and Alfons Juan. 2019. The MLLP-UPV Spanish-Portuguese and Portuguese-Spanish Machine Translation Systems for WMT19 Similar Language Translation Task. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 181–186, Florence, Italy. Association for Computational Linguistics.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

Hideki Isozaki, Tsutomu Hirao, Kevin Duh, Katsuhito Sudoh, and Hajime Tsukada. 2010. Automatic evaluation of translation quality for distant language pairs. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 944–952, Cambridge, MA. Association for Computational Linguistics.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Vancouver, Canada. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, Greg S. Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *In Proceedings of Association for Machine Translation in the Americas*, pages 223–231.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

NMT based Similar Language Translation for Hindi - Marathi

Vandan Mujadia and Dipti Misra Sharma
Machine Translation - Natural Language Processing Lab
Language Technologies Research Centre
Kohli Center on Intelligent Systems
International Institute of Information Technology - Hyderabad
vandan.mu@research.iiit.ac.in, dipti@iiit.ac.in

Abstract

This paper describes the participation of team F1toF6 (LTRC, IIIT-Hyderabad) for the WMT 2020 task, similar language translation. We experimented with attention based recurrent neural network architecture (seq2seq) for this task. We explored the use of different linguistic features like POS and Morph along with back translation for Hindi-Marathi and Marathi-Hindi machine translation.

1 Introduction

Machine Translation (MT) is the field of Natural Language Processing which aims to translate a text from one natural language (i.e Hindi) to another (i.e Marathi). The meaning of the resulting translated text must be fully preserved as the source text in the target language.

For the translation task, different types of machine translation systems have been developed and they are mainly Rule based Machine Translation (RBMT)(Forcada et al., 2011), Statistical Machine Translation (SMT) (Koehn, 2009) and Neural Machine Translation (NMT) (Bahdanau et al., 2014).

Statistical Machine Translation (SMT) aims to learn a statistical model to determine the correspondence between a word from the source language and a word from the target language. Neural Machine Translation is an end to end approach for automatic machine translation without heavily hand crafted feature engineering. Due to recent advances, NMT has been receiving heavy attention and achieved state of the art performance in the task of language translation. With this work, we intend to check how NMT systems could be used for low resource and similar language machine translation.

Data	Sents	Token	Type
Hindi (Parallel)	38,246	7.6M	39K
Marathi (Parallel)	38,246	5.6M	66K
Hindi (Mono)	80M	-	-
Marathi (Mono)	3.2M	-	-

Table 1: Hindi-Marathi WMT2020 Training data

This paper describes our experiments for the task of similar language translation of WMT-2020. We focused only on Hindi-Marathi language pair for the translation task (both directions). The origin of these two languages are the same as they are Indo-aryan languages(wikipedia, 2020). Hindi is said to have evolved from Sauraseni Prakrit (wikipedia Hindi, 2020) whereas Marathi is said to have evolved from Maharashtri Prakrit (wikipedia Marathi, 2020). They also have evolved as two major languages in different regions of India.

In this work, we focused only on recurrent neural network with attention based sequence to sequence architecture throughout all experiments. Along with it, we also explored the morph(Virpioja et al., 2013) induced sub-word segmentation with byte pair encoding (BPE)(Sennrich et al., 2016b) to enable open vocabulary translation. We used POS tags as linguistic feature and back translation to leverage synthetic data for machine translation task in both directions. In the similar language translation task of WMT-2020, we participated as team named “f1plusf6”.

2 Data

We utilised parallel and monolingual corpora provided for the task on Hindi<->Marathi language pairs. Table-1 describes the training data (parallel

Proceedings of the 5th Conference on Machine Translation (WMT), pages 414–417

and monolingual) on which we carried out all experiments. We deliberately excluded Indic WordNet data from the training after doing manual quality check. As this is a constrained task, our experiments do not utilise any other available data.

3 Pre-Processing

As a first pre-precessing step we use IndicNLP Toolkit[1] along with an in-house tokenizer to tokenize and clean both Hindi and Marathi corpora (train, test, dev and monolingual).

3.1 Morph + BPE Segmentation

Marathi and Hindi are morphologically rich languages and from the Table-1, based on the the comparative token/type ratio, one can find that Marathi is a more agglutinative language than Hindi. Translating from morphologically-rich agglutinative languages is more difficult due to their complex morphology and large vocabulary. To address this issue, we have come up with a segmentation method which is based on morph and BPE segmentation (Sennrich et al., 2016b) as a pre-processing step.

In this method, we utilised unsupervised Morfessor (Virpioja et al., 2013) to train a Morfessor model on monolingual data for both languages. We then applied this trained Morfessor model on our corpora (train, test, validation) to get meaningful stem, morpheme, suffix segmented sub-tokens for each word in each sentence.

(1) aur jab maansaahaaree
pakshee lothon par jhapate ,
tab abraam ne unhen uda diya .
'And when the carnivorous birds swooped on the carcasses, Abram blew them away.'

(2) aur jab maansaa##haaree
pakshee loth##on par jhapat##e ,
tab ab##raam ne unhen uda diya .
'And when the carnivorous birds swooped on the carcasses, Abram blew them away.'

(3) aur jab maan@@ saa##haaree
pakshee loth##on par jha@@ pat##e ,
tab ab##raam ne unhen uda diya .
'And when the carnivorous birds swooped on the carcasses, Abram blew them away.'

We demonstrate this method with a Hindi sentence as given in Example-1. Example -1, shows Hindi text with romanized text and the corresponding English translation for better understanding. The Example-2 shows the same sentence with Morfessor based segmentation with token ##. Here we notice that Morfessor model has segmented the Hindi words into meaningful stems and suffixes. i.e maansaahaaree=maansaa + haaree(meat + who eats). We would like to use it in our experiments to tackle the difficulties that arise due to complex morphology at the source language in machine translation tasks. On top of this morph segmented text we applied BPE (Sennrich et al., 2016a) as given in Example-3. Here @@ is sub-word separator for byte pair based segmentation and ## is the separator for morph based segmentation.

3.2 Features

For Hindi to Marathi translation, we carried out experiments using Part of Speech (POS) tags as a word level as well as a subword level feature as described in (Sennrich and Haddow, 2016). We use LTRC shallow parser[2] toolkit to get POS tags.

4 Training Configuration

Recurrent Neural Network (RNN) based machine translation models work on encoder-decoder based architecture. Here, the encoder takes the input (source sentence) and encodes it into a single vector (called as a context vector). Then the decoder takes this context vector to generate an output sequence (target sentence) by generating a word at a time(Sutskever et al., 2014). Attention mechanism is an extension to this sequence to sequence architecture to avoid attempting to learn a single vector. Instead, based on learnt attention weights, it focuses more on specific words at the source end and generates a word at a time. More details can be found here (Bahdanau et al., 2014), (Luong et al., 2015).

For our experiments, we utilize sequence to sequence NMT model with attention for all of our experiments with following configuration.

[1] http://anoopkunchukuttan.github.io/indic nlp library/

[2] http://ltrc.iiit.ac.in/analyzer/

Model	Feature	BPE (Merge ops)	BLEU
BiLSTM + LuongAttn	Word level	-	19.70
BiLSTM + LuongAttn	Word + Shared Vocab (SV)+ POS	-	20.49
BiLSTM + LuongAttn	BPE	10K	20.1
BiLSTM + LuongAttn	BPE+SV+MORPH Segmentation	10K	20.44
BiLSTM + LuongAttn	BPE+SV+MORPH+POS	10K	**20.62**
BiLSTM + LuongAttn	BPE+SV+MORPH+POS + BT	10K	16.49

Table 2: BLEU scores on Development data for Hindi-Marathi

Model	Feature	BPE (Merge ops)	BLEU
BiLSTM + LuongAttn	Word level	-	21.42
BiLSTM + LuongAttn	Word + Shared Vocab (SV)	-	23.84
BiLSTM + LuongAttn	BPE	20K	24.56
BiLSTM + LuongAttn	BPE+SV+MORPH Segmentation	20K	25.36
BiLSTM + LuongAttn	BPE+SV+MORPH+POS	20K	**25.55**
BiLSTM + LuongAttn	BPE+SV+MORPH+POS + BT	20K	23.80

Table 3: BLEU scores on Development data for Marathi-Hindi

- Morph + BPE based subword segmentation, POS tags as feature
- Embedding size : 500
- RNN for encoder and decoder: bi-LSTM
- Bi-LSTM dimension : 500
- encoder - decoder layers : 2
- Attention : luong (general)
- copy attention(Gu et al., 2016) on dynamically generated dictionary
- label smoothing : 1.0
- dropout : 0.30
- Optimizer : Adam
- Beam size : 4 (train) and 10 (test)

As these are two similar languages, share writing scripts and large sets of named entities, we used shared vocab across training. We used Opennmt-py (Klein et al., 2020) toolkit with above configuration for our experiments.

5 Back Translation

Back translation is a widely used data augmentation method for low resource neural machine translation(Sennrich et al., 2016a). We utilised monolingual data (i.e of Marathi) and a NMT model trained on given training data for a direction (i.e, Marathi to Hindi) to enrich training data of the opposite directional NMT training (i.e, Hindi - Marathi) by populating synthetic data. We used around 5M back translated pairs (after perplexity based pruning with respect to sentence length) for both translation directions.

Using above described configuration, we performed experiments based on different parameter (feature) configurations. We trained and tested our models on word level, BPE level and morph + BPE level for input and output. We also used POS tagger and experimented with shared vocabulary across the translation task. The results are discussed in following Result section.

6 Result

Table-2 and Table-3 show performance of systems with different configuration in terms of BLEU score(Papineni et al., 2002) for Hindi-Marathi and Marathi-Hindi respectively on the validation data. We achieved 20.62 and 25.55 development and 5.94 and 18.14 test BLEU scores for Hindi-Marathi and Marathi-Hindi systems respectively.

The results show that for low resource similar language settings, MT models based on sequence to sequence neural network can be improved with linguistic information like morph based segmentation and POS features. The results also show that morph based segmentation along with

byte pair encoding improves BLEU score for both directions. But Marathi-Hindi directed translation shows considerable improvement. Therefore our method shows improvement while translating from morphologically richer language (Marathi) to comparatively less morphologically richer language (Hindi).

The results also suggest that the use of back translated synthetic data for low resource language pairs reduces the overall performance marginally. The reason for this could be, due to low quantity of training data for NMT models, they could be over learning and back translation could be helping to do better generalization.

7 Conclusion

We conclude from our experiments that linguistic feature driven NMT for similar low resource languages is a promising approach. We also believe that morph+BPE based segmentation is a potential segmentation method for morphologically richer languages.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Mikel L Forcada, Mireia Ginestí-Rosell, Jacob Nordfalk, Jim O'Regan, Sergio Ortiz-Rojas, Juan Antonio Pérez-Ortiz, Felipe Sánchez-Martínez, Gema Ramírez-Sánchez, and Francis M Tyers. 2011. Apertium: a free/open-source platform for rule-based machine translation. *Machine translation*, 25(2):127–144.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1631–1640.

wikipedia Hindi. 2020. Shauraseni prakrit - wikipedia. https://en.wikipedia.org/wiki/Shauraseni_Prakrit. (Accessed on 08/15/2020).

Guillaume Klein, François Hernandez, Vincent Nguyen, and Jean Senellart. 2020. The opennmt neural machine translation toolkit: 2020 edition. In *Proceedings of the 14th Conference of the Association for Machine Translation in the Americas (AMTA 2020)*, pages 102–109.

Philipp Koehn. 2009. *Statistical machine translation*. Cambridge University Press.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421.

wikipedia Marathi. 2020. Maharashtri prakrit - wikipedia. https://en.wikipedia.org/wiki/Maharashtri_Prakrit. (Accessed on 08/15/2020).

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Rico Sennrich and Barry Haddow. 2016. Linguistic input features improve neural machine translation. In *Proceedings of the First Conference on Machine Translation: Volume 1, Research Papers*, pages 83–91.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Sami Virpioja, Peter Smit, Stig-Arne Grönroos, Mikko Kurimo, et al. 2013. Morfessor 2.0: Python implementation and extensions for morfessor baseline.

wikipedia. 2020. Indo-aryan languages - wikipedia. https://en.wikipedia.org/wiki/Indo-Aryan_languages. (Accessed on 08/17/2020).

NUIG-Panlingua-KMI Hindi ↔Marathi MT Systems for Similar Language Translation Task @ WMT 2020

Atul Kr. Ojha[!+], Priya Rani[!], Akanksha Bansal[+], Bharathi Raja Chakravarthi[!], Ritesh Kumar[*], John P. McCrae[!]

[!]Data Science Institute, NUIG, Galway, [+]Panlingua Language Processing LLP, New Delhi, [*]Dr. Bhimrao Ambedkar University, Agra

(atulkumar.ojha,priya.rani,bharathi.raja)@insight-centre.org, panlingua@outlook.com, john.mccrae@nuigalway.ie, ritesh78_llh@jnu.ac.in

Abstract

NUIG-Panlingua-KMI submission to WMT 2020 seeks to push the state-of-the-art in the Similar language translation task for the Hindi ↔ Marathi language pair. As part of these efforts, we conducted a series of experiments to address the challenges for translation between similar languages. Among the 4 MT systems prepared for this task, 1 PBSMT systems were prepared for Hindi ↔ Marathi each and 1 NMT systems were developed for Hindi ↔ Marathi using Byte Pair Encoding (BPE) of subwords. The results show that different architectures in NMT could be an effective method for developing MT systems for closely related languages. Our Hindi-Marathi NMT system was ranked 8th among the 14 teams that participated and our Marathi-Hindi NMT system was ranked 8th among the 11 teams participated for the task.

1 Introduction

Developing automated relations between closely related languages is a contemporary concern especially in the domain of Machine Translation(MT). Hindi and Marathi exhibit a significant overlap in their vocabularies and strong syntactic plus lexical similarities. These striking similarities seem promising in enhancing the possibility of mutual inter-comprehension within closely related languages. However, automated translation between such closely related languages is a rather challenging task.

The linguistic similarities and regularities in morphological variations and orthography motivate the use of character-level translation models, which have been applied to translation (Vilar et al., 2007; Chakravarthi et al., 2020) and transliteration (Matthews, 2007; Chakravarthi et al., 2019a; Chakravarthi, 2020). In the past few years, neural machine translation systems have achieved outstanding performance with high resource languages, with the help of open source toolkit such as OpenNMT (Klein et al., 2017), Marian (Junczys-Dowmunt et al., 2018) and Neamtus (Sennrich et al., 2017), which provide various ways of experimenting with the use of different features and architectures, yet it fails to achieve the same results with low resource languages (Chakravarthi et al., 2018, 2019b). However, Sennrich and Zhang (2019) revisited the NMT models and tuned hyper-parameters, changed network architectures to optimize NMT for low-resource conditions and concluded that low-resource NMT is very sensitive to hyper-parameters such as Byte Pair Encoding (BPE) vocabulary size, word dropout, and others. This paper is an extension of our work Ojha et al. (2019) submitted to WMT 2019 similar language translation task. Therefore our team adapted methods of the low resource setting for NMT proposed by Sennrich and Zhang (2019) to explore the following broad objectives:

- to compare the performance of SMT and NMT in case of closely related, relatively low-resourced language pairs, and
- to findout how to leverage the accuracy of NMT in closely related languages using BPE into subwords.
- to analyze the effects of data quality in performance of the systems.

2 System Description

This section provides an overview of the systems developed for the WMT 2020 Shared Task. In these experiments, the NUIG-Panlingua-KMI team explored two different approaches: phrase-based statistical (Koehn et al., 2003), and neural method for Hindi-Marathi and Marathi-Hindi language pairs. In all the submitted systems, we use the Moses (Koehn et al., 2007) and Nematus (Sennrich et al., 2017) toolkit for developing statistical and neural

Proceedings of the 5th Conference on Machine Translation (WMT), pages 418–423
Online, November 19–20, 2020.

machine translation systems respectively. The pre-processing was done to handle noise in data (for example, different language sentences, non-UTF characters etc), the details of which are provided in section 3.1

2.1 Phrase-based SMT Systems

These systems were built on the Moses open source toolkit using the KenLM (Heafield, 2011) language model and GIZA++ (Och and Ney, 2003) aligner. 'Grow-diag-final-and heuristic' parameters were used to extract phrases from the corresponding parallel corpora. In addition to this, KenLM was used to build 5-gram language models.

2.2 Neural Machine Translation System

Nematus was used to build 2 NMT systems. As we mentioned in an earlier section, at first data was pre-processed at subwords level with BPE for neural translation, and then the system was trained using Nematus toolkit. Most of the system features were adopted from (Sennrich et al., 2017; Koehn and Knowles, 2017) (see section 3.3.2).

2.3 Assessment

Assessment of these systems was done on the standard automatic evaluation metrics: BLEU (Papineni et al., 2002), Rank-based Intuitive Bilingual Evaluation Score (RIBES) (Isozaki et al., 2010) and Translation Error Rate (TER) (Snover et al., 2006).

3 Experiments

This section briefly describes the experiment settings for developing the systems.

3.1 Data Preparations

The parallel data-set for these experiments was provided by the *WMT Similar Translation Shared Task* [1] organisers and the Marathi monolingual data-set was taken from *WMT 2020 Shared Task: Parallel Corpus Filtering for Low-Resource Conditions*.[2] The parallel data was sub-divided into training, tuning, and monolingual sets, as detailed in Table 1. However, the shared data was very noisy.

To enhance the data quality, the team had to undertake an extensive pre-processing session focused on identifying and cleaning the data-sets. Out of 43274 training sentences, the Hindi corpus had Telugu sentences while the Marathi corpus had Meitei sentences intermingled as shown in first row (Figure 1). The parallel data had more than 1192 lines that were not comparable with each other as shown in second and third row (Figure 1), where some Hindi sentences had only half the sentences translated in Marathi (second row) and some had blank spaces against their Marathi counter parts (third row). The translation quality of the parallel data was also not up to mark. In fact, the team could locate a few instances of synthetic data. There were a few sentences where character encoding was an issue, hence were completely unintelligible.

Language Pair	Training	Tuning	Monolingual
Hindi ↔ Marathi	43274	1411	-
Marathi	-	-	326748
Hindi	-	-	75348193

Table 1: Statistics of Parallel and Monolingual Sentences of the Hindi and Marathi Languages

3.2 Pre-processing

The following pre-processing steps were performed as part of the experiments:

a) Both corpora were tokenized and cleaned (sentences of length over 80 words were removed).

b) For neural translation, training, validation and test data was prepossessed into subwords BPE format. This format was utilised to prepare BPE and vocabulary further used.

All these processes were performed using Moses scripts. However, the tokenization was done by the RGNLP team tokenizer (Ojha et al., 2018) and Indic_nlp_library.[3] These tokenizers were used since Moses does not provide a tokenizer for Indic languages. Also the RGNLP tokenizer ensured that the canonical Unicode representation of the characters are retained.

3.3 Development of the NUIG-Panlingua-KMI MT Systems

After removing noisy and pre-processing data, the following steps were followed to build the NUIG-Panlingua-KMI MT systems:

[1] http://www.statmt.org/wmt20/similar.html

[2] https://wmt20similar.cs.upc.edu/

[3] https://github.com/anoopkunchukuttan/indic_nlp_library

Sr.No	Hindi Sentences	Marathi Sentences
1	ಬಾಳೆಯಣ್ಣು , ಪೊಪ್ಪಾಯ , ಸೀತಾಫಲ ಮುಂತಾದವುಗಳು ಶೀಣ ಗುಣ ಹೊಂದಿದ್ದರೆ ಶುಂಟಿ , ಬೆಳ್ಳುಳ್ಳಿ , ಈರುಳ್ಳಿ ಮುಂತಾದವು ಉಷ್ಣ ಗುಣವನ್ನು ಹೊಂದಿರ	[illegible]
2	सचिन तेंदुलकर ने एकदिवसी क्रिकेट में सर्वाधिक शतक बनाकर , क्रिकेट जगत में अपनी प्रधानता सिद्ध कर दी । / आधुनिक युग में वैज्ञानिकों की प्रधानता को झुठलाया	आजच्या शिक्षणपद्धतीत विज्ञान विषयाला प्राधान्य आहे ।
3	अप्रियभाषी व्यक्ति निंदा का पात्र होता है। / अप्रियभाषी व्यक्ति अपनी बोली से अपनों में भी पराया बना रहता है ।	-
4	हर देशवासी को हमारी इस मातृ-शक्ति पर, हमारी इन वीरांगनाओं के प्रति आदर होना बहुत स्वाभाविक है। मैं इन दोनों बहनों को हृदय से बहुत-बहुत बधाई देता हू।	प्रत्येक देशवासियाला आपल्या या मातृ-शक्ती विषयी, आपल्या या वीरांगनांविषयी आदर वाटणे स्वाभाविक आहे.

Figure 1: Examples of discrepancies in Hindi-Marathi parallel data

Example-1	
Source	अपने संबोधन के दौरान, प्रधानमंत्री ने समारोह में उपस्थित व्यक्तियों को, विशेषकर के जो पहले भारत नहीं आ चुके हैं, उन्हें भारत की सांस्कृतिक एवं आध्यात्मिक परंपरा का अनुभव करने के लिये जनवरी 2019 में आयोजित होने वाले कुंभ मेले में आने के लिये आमंत्रित भी किया।
Reference	आपल्या भाषणादरम्यान, पंतप्रधानांनी भारतातील सांस्कृतिक आणि आध्यात्मिक परंपरेचा अनुभव घेण्यासाठी जानेवारी 201 9 मध्ये आयोजित कुंभमेळ्याच्या निमित्ताने भारताला अजून पर्यंत भेट न दिलेल्या लोकांना उपस्थित राहण्याचे आवाहन केले.
NUIG-Panlingua-KMI_P	यावेळी दौरान, पंतप्रधानांनी समारंभात उपस्थित व्यक्ती को, विशेषत : जे आधी भारत नाही आले हैं, त्यांना भारतातील सांस्कृतिक आणि आध्यात्मिक परंपरेचा अनुभव करण्यासाठी जानेवारी 2019 मधील होणाऱ्या कुंभमेळ्यात येण्यासाठी निमंत्रण दिले .
NUIG-Panlingua-KMI_C	बोलतांना दौरान, पंतप्रधानांनी दिलेल्या को, विशेषत : जे आधी भारत नाही आलो हैं, भारतातील सांस्कृतिक आणि आध्यात्मिक परंपरेचा अनुभव घेण्यासाठी जानेवारी 201 9 मध्ये आयोजित कुंभमेळ्याच्या निमित्ताने भारताला अजून पर्यंत भेट न .
Example-2	
Source	आमचा राष्ट्राला अर्पण केलेला पश्चिम मालवाहतूक कॉरिडॉर आता पूर्ण होणार आहे.
Reference	हमारा वेस्टर्न डेडिकेटेड फरेट कॉरिडोर अब पूरा होने वाला है ।
NUIG-Panlingua-KMI_P	हमारा राष्ट्र को समर्पित किया पश्चिम कार्गो कॉरिडोर अब पूरी हो गई है ।
NUIG-Panlingua-KMI_C	हमारा राष्ट्र को समर्पित करने के पश्चिम माल कॉरिडोर अब लगभग पूरा होने वाला है ।
Example-3	
Source	उन्होंने शौचालय की प्रतिष्ठा बढ़ा दी है।
Reference	त्यांनी शौचालयाची प्रतिष्ठा वाढवली आहे.
NUIG-Panlingua-KMI_P	त्यांनी शौचालय यांच्या प्रतिष्ठा वाढवली आहे .
NUIG-Panlingua-KMI_C	त्यांनी शौचालयांना प्रतिष्ठा वाढवली आहे .

Figure 2: Analysis of the PBSMT and NMT's Systems

3.3.1 Building Primary MT Systems:

As previously mentioned, the Hindi-Marathi and Marathi-Hindi PBSMT systems were built as the primary submission using Moses. The language model was built first, using KenLM. For Marathi-Hindi and Hindi-Marathi language pairs, the language models were trained on 5-gram. After that, the systems were built independently and combined in a loglinear scheme in which each model was assigned a different weight using the Minimum Error Rate Training (Och, 2003) tuning algorithm. To train and tune the systems, we used 40454 and 1411

parallel sentences, respectively, for all language pairs.

3.3.2 Building Contrastive MT Systems:

As mentioned in the previous section, Nematus toolkit was used to develop the NMT systems. The training was done on subword and character-level. All the NMT experiments were carried out only with a data-set that contained sentences with length of up to 80 words. The neural model is trained on 5000 epochs, using Adam with a default learning rate of 0.002, dropout at 0.01 and mini-batches of 80 and the batch size for the validation was 40. Vocabulary size of 30000 for both Marathi-Hindi and Hindi-Marathi language pairs was extracted. Remaining parameters were limited with the use of default hyper-parameters configuration.

4 Evaluation

All the systems were evaluated using the reference set provided by the shared task organizers. The standard MT evaluation metrics, BLEU (Papineni et al., 2002) score, RIBES (Isozaki et al., 2010) and TER (Snover et al., 2006), were used for automatic evaluation. These results were prepared on the Primary and Contrastive system submission which are mentioned in the Table 2 as _P and _C, where *_P* stands for Primary and *_C* stands for Contrastive, respectively. It gives a quantitative picture of particular differences across different systems, especially with reference to evaluation scores (Table 2)

System	BLEU	RIBES	TER
Hindi-Marathi_P	9.38	51.88	91.24
Hindi-Marathi_C	9.76	52.18	91.49
Marathi-Hindi_P	17.38	59.31	81.47
Marathi-Hindi_C	17.39	58.84	81.15

Table 2: Accuracy of Hindi↔Marathi MT Systems at BLEU, RIBES and TER Metrics

4.1 Results

Overall we see varying performance among the system submitted to the task, with some performing much better out-of-sample than others. The NUIG-Panlingua-KMI subword NMT system took 8th position for both Hindi-Marathi and Marathi-Hindi language pair, across 14 teams. Our subword NMT systems for Marathi-Hindi language pair showed better results in terms of all the three metrics (17.39 in BLEU, 58.84 in RIBES and 81.15 in TER) while the Hindi-Marathi language pair scored 9.76 in BLEU, 52.18 in RIBES and 91.24 in TER. Across both the language pairs, subword based NMT performed better than PBSMT as its accuracy rate was higher in BLEU and lower in TER metrics, shown in Table 2.

4.2 Analysis

We used the reference set provided by the shared task organizers to evaluate both PBSMT and NMT systems. Even though subword based NMT system could take advantage of the shared features among similar languages, challenges in translating a few linguistics structures acted as a constraint. Example 1 shown in Figure 2 is one of the challenging structures that the system was unable to translate. In these sentences the systems could not capture the correct tense and aspect which is past perfect in source sentence whereas the NMT system translated it as simple past. The second most common challenging structures that needed special attention were the postpositions as shown in Example 2 and 3 in the figure. In most cases, the system over-generalised the sentences in Marathi and generated unnecessary postposition phrases in Hindi as in Example 2. Similarly, we can see in Example 3 while translating from Hindi to Marathi both PBSMT and NMT systems used wrong post-positions.

5 Conclusion

Our experiment results reveal that subword based NMT could take advantage of the relation between the similar language to boost the accuracy of neural machine translations system in low resource data settings. As BPE units are variable-length units and the vocabularies used are much smaller than morpheme and word-level model, the problem of data sparsity does not occur. On the contrary, it provides an appropriate context for translation between similar languages. However, the quality of data used to train the systems does affect the quality of translation. Thus, we could conclude that shared features between two languages could be an advantage to leverage the accuracy of NMT systems for closely related languages.

Acknowledgments

This publication has emanated from research in part supported by the Irish Research Council under grant number SFI/18/CRT/6223 (CRT-Centre for Research Training in Artificial Intelligence) co-funded by the European Regional Development Fund as well as by the EU H2020 programme un-

der grant agreements 731015 (ELEXIS-European Lexical Infrastructure).
We are also grateful to the organizers of WMT Similar Translation Shared Task 2020 for providing us the Hindi↔Marathi Parallel Corpus, monolingual and evaluation scores.

References

Bharathi Raja Chakravarthi. 2020. *Leveraging orthographic information to improve machine translation of under-resourced languages*. Ph.D. thesis, NUI Galway.

Bharathi Raja Chakravarthi, Mihael Arcan, and John P McCrae. 2018. Improving wordnets for under-resourced languages using machine translation. In *Proceedings of the 9th Global WordNet Conference (GWC 2018)*, page 78.

Bharathi Raja Chakravarthi, Mihael Arcan, and John P McCrae. 2019a. Comparison of different orthographies for machine translation of under-resourced Dravidian languages. In *2nd Conference on Language, Data and Knowledge (LDK 2019)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

Bharathi Raja Chakravarthi, Mihael Arcan, and John Philip McCrae. 2019b. Wordnet gloss translation for under-resourced languages using multilingual neural machine translation. In *Proceedings of the Second Workshop on Multilingualism at the Intersection of Knowledge Bases and Machine Translation*, pages 1–7.

Bharathi Raja Chakravarthi, Priya Rani, Mihael Arcan, and John P McCrae. 2020. A survey of orthographic information in machine translation. *arXiv e-prints*, pages arXiv–2008.

Kenneth Heafield. 2011. Kenlm: Faster and smaller language model queries. In *Proceedings of the sixth workshop on statistical machine translation*, pages 187–197. Association for Computational Linguistics.

Hideki Isozaki, Tsutomu Hirao, Kevin Duh, Katsuhito Sudoh, and Hajime Tsukada. 2010. Automatic evaluation of translation quality for distant language pairs. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 944–952.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia. Association for Computational Linguistics.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M Rush. 2017. Opennmt: Open-source toolkit for neural machine translation. *arXiv preprint arXiv:1701.02810*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39, Vancouver. Association for Computational Linguistics.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 48–54. Association for Computational Linguistics.

David Matthews. 2007. Machine transliteration of proper names. *Master's Thesis, University of Edinburgh, Edinburgh, United Kingdom*.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 160–167, Sapporo, Japan. Association for Computational Linguistics.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Atul Kr Ojha, Koel Dutta Chowdhury, Chao-Hong Liu, and Karan Saxena. 2018. The RGNLP machine translation systems for WAT 2018. In *Proceedings of the 5th Workshop on Asian Translation (WAT2018)*.

Atul Kr Ojha, Ritesh Kumar, Akanksha Bansal, and Priya Rani. 2019. Panlingua-KMI MT system for similar language translation task at WMT 2019. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 213–218.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Rico Sennrich, Orhan Firat, Kyunghyun Cho, Alexandra Birch, Barry Haddow, Julian Hitschler, Marcin Junczys-Dowmunt, Samuel Läubli, Antonio Valerio Miceli Barone, Jozef Mokry, et al. 2017. Nematus: a toolkit for neural machine translation. *arXiv preprint arXiv:1703.04357*.

Rico Sennrich and Biao Zhang. 2019. Revisiting low-resource neural machine translation: A case study. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 211–221, Florence, Italy. Association for Computational Linguistics.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of association for machine translation in the Americas*, volume 200.

David Vilar, Jan-Thorsten Peter, and Hermann Ney. 2007. Can we translate letters? In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 33–39, Prague, Czech Republic. Association for Computational Linguistics.

Neural Machine Translation for Similar Languages: The Case of Indo-Aryan Languages

Santanu Pal[1], Marcos Zampieri[2]
[1]Wipro AI Lab, India
[2]Rochester Institute of Technology, USA
santanu.pal2@wipro.com

Abstract

In this paper we present the WIPRO-RIT systems submitted to the Similar Language Translation shared task at WMT 2020. The second edition of this shared task featured parallel data from pairs/groups of similar languages from three different language families: Indo-Aryan languages (Hindi and Marathi), Romance languages (Catalan, Portuguese, and Spanish), and South Slavic Languages (Croatian, Serbian, and Slovene). We report the results obtained by our systems in translating from Hindi to Marathi and from Marathi to Hindi. WIPRO-RIT achieved competitive performance ranking 1st in Marathi to Hindi and 2nd in Hindi to Marathi translation among 22 systems.

1 Introduction

WMT 2020 is the fifth edition of WMT as a conference following a series of well-attended workshops that date back to 2006. WMT became a well-established conference due to its blend of research papers and popular shared tasks on different topics such as translation in various domains (e.g. biomedical, news), translation quality estimation, and automatic post-editing. The competitions co-organized with WMT provide important datasets and benchmarks widely used in the MT community. The vast majority of these tasks so far, however, involved training systems to translate to and from English (Bojar et al., 2016, 2017) while only a few of them addressed the problem of translating between pairs of languages with less resources.

To address this issue, in 2019, the Similar Language Translation (SLT) shared task was introduced at WMT. SLT's purpose was to evaluate the performance of state-of-the-art MT systems on translating between pairs of similar languages without English as a pivot language (Barrault et al., 2019). The organizers provided participants with training, development, and testing parallel data from three pairs of languages from three different language families: Spanish - Portuguese (Romance languages), Czech - Polish (Slavic languages), and Hindi - Nepali (Indo-Aryan languages). Systems were evaluated using automatic metrics, namely BLEU (Papineni et al., 2002) and TER (Snover et al., 2006).

In SLT 2020, the task organizes once again included an Indo-Aryan language track with Hindi and Marathi. Indo-Aryan languages are a sub-family of the Indo-European language family which includes Bengali, Bohjpuri, Hindi, Marathi, and Nepali. These languages are mainly spoken in North and Central India, and some neighbouring countries such as Nepal, Bangladesh, and Pakistan etc. The script used in most of these languages are derived from the ancient Brahmi script and enriched with high grapheme to phoneme correspondence leading to many orthographic similarities across these languages.

In addition to Hindi and Marathi, SLT 2020 features two other tracks with similar languages from the following language families: Romance languages (Catalan, Portuguese, and Spanish) and South Slavic Languages (Croatian, Serbian, and Slovene). In this paper we describe the WIPRO-RIT submission to the SLT 2020 Indo-Aryan track. Our WIPRO-RIT system is based on the model described in Johnson et al. (2017). WIPRO-RIT achieved competitive performance ranking 1st in Marathi to Hindi and 2nd in Hindi to Marathi translation among 22 systems.

Proceedings of the 5th Conference on Machine Translation (WMT), pages 424–429

2 Related Work

With the substantial performance improvements brought to MT by neural approaches, a growing interest in translating between pairs of similar languages, language varieties, and dialects has been observed. Recent studies have addressed MT between Arabic dialects (Harrat et al., 2019; Shapiro and Duh, 2019) Catalan and Spanish, Croatian and Serbian (Popović et al., 2020), (Costa-jussà, 2017), Brazilian and European Portuguese (Costa-jussà et al., 2018), and several pairs of languages and language varieties such as Brazilian and European Portuguese, Canadian and European French, and similar languages such as Croatian and Serbian, and Indonesian and Malay (Lakew et al., 2018).

The interest on diatopic language variation is evidenced by the recent iterations of the VarDial workshop in which papers on MT applied to similar languages varieties, and dialects (Shapiro and Duh, 2019; Myint Oo et al., 2019; Popović et al., 2020) have been presented along with evaluation campaigns featuring multiple shared tasks on a number of related topics such as cross-lingual morphological analysis, cross-lingual parsing, dialect identification, and morphosyntactic tagging (Zampieri et al., 2018, 2019; Găman et al., 2020).

3 Data

For our experiments, we use the Hindi–Marathi and Marathi–Hindi WMT 2020 SLT data. The released parallel dataset was collected from news (Siripragada et al., 2020), PMIndia (Haddow and Kirefu, 2020) and Indic Wordnet (Bhattacharyya, 2010; Kunchukuttan, 2020a) datasets. To augment our dataset, we use English–Hindi parallel data released in WMT 2014 (Bojar et al., 2014), consisting of more than 2 million parallel sentences, which is available as an additional resource. We use a subset of 5 million segments of Hindi monolingual news crawled from ca. 32 million data. We also use a subset 5 million Marathi monolingual data. We performed similar cleaning and pre-processing methods as we described in case of parallel data.

The five million Hindi monolingual sentences were first back-translated to English using a Hindi–English NMT system. The Hindi–English NMT system was trained on English–Hindi parallel data released in WMT 2014 (Bojar et al., 2014), IITB parallel corpus (Kunchukuttan et al., 2018), the parallel dataset was collected from news (Siripragada et al., 2020) and the PMIndia (Haddow and Kirefu, 2020) parallel corpus (see Table 1).

Data Sources	#sentences
WMT	273,885
News	156,344
IITB	1,561,840
PM India	56,831
Total	2,048,900
Remove duplicates	1,464,419
Cleaning*	961,036

Table 1: English–Hindi parallel data statistics. *Removing noisy mixed language sentences.

We also back-translated 5 million Marathi monolingual sehments using our WIPRO-RIT CONTRASTIVE 1 system described in more detail Section 6. For Marathi–Hindi we did not use any back translation data in our CONTRASTIVE 2 and PRIMARY submissions. In the both cases 5 million English–Hindi back-translation data provide significant ($p < 0.01$) improvements over CONTRASTIVE 1 (detailed in Section 6).

The released WMT 2014 EN-HI data and the WMT SLT 2020 data were noisy for our purposes, so we apply methods for cleaning (see data statistics in Table 2).

Parallel	#sentences
News	12,349
PM India	25,897
Indic WordNet	11,188
Total	49,434
Filtered*	33923

Table 2: Data statistics of released SLT Data; *Filtration methods: (i) remove duplicates and (ii) filtering noisy mixed language sentences.

We performed the following two steps: (i) we use the cleaning process described in Pal et al. (2015), and (ii) we execute the Moses (Koehn et al., 2007) corpus cleaning scripts with minimum and maximum number of tokens set to 1 and 100, respectively. After cleaning and re-

L1 → L2		Parallel Sentences	
		Source	Target
HI→MR	Raw data	देश एकल प्रयासों से आगे बढ़ चुके हैं।	देश आता सामाईक प्रयत्न करत आहेत.
	Processed data	TO_MR देश एकल प्रयासों से आगे बढ़ चुके हैं।	देश आता सामाईक प्रयत्न करत आहेत.
MR→HI	Raw data	देश आता सामाईक प्रयत्न करत आहेत.	देश एकल प्रयासों से आगे बढ़ चुके हैं।
	Processed data	TO_HI देश आता सामाईक प्रयत्न करत आहेत.	देश एकल प्रयासों से आगे बढ़ चुके हैं।
EN→HI	Raw data	The MoU was signed in February, 2016.	इस एमओयू पर फरवरी, 2016 में हस्ताक्षर किए गए थे।
	Processed data	TO_HI The MoU was signed in February, 2016.	इस एमओयू पर फरवरी, 2016 में हस्ताक्षर किए गए थे।

Table 3: Multilingual **Processed data**, indicating TO_XX as target language:

moving duplicates, we have 1M EN-HI parallel sentences. Next, we perform punctuation normalization, and then we use the Moses tokenizer to tokenize the English side of the parallel corpus with 'no-escape' option. Finally, we apply true-casing. For the case of Hindi and Marathi, we use Indic NLP Library[1] (Kunchukuttan, 2020b) for tokenizaton.

4 Model Architecture

Our model is based on a transformer architecture (Vaswani et al., 2017) built solely upon such attention mechanisms completely replacing recurrence and convolutions. The transformer uses positional encoding to encode the input and output sequences, and computes both self- and cross-attention through so-called multi-head attentions, which are facilitated by parallelization. We use multi-head attention to jointly attend to information at different positions from different representation subspaces.

We present a single multilingual NMT system based on the transformer architecture that can translate between multiple languages. To make use of multilingual data within a single NMT model, we perform one simple modification to the source side of the multilingual data, we use an additional token at the beginning of the each source sentence to indicate the target language by the NMT model would be translated as shown in Table 3.

We train the model with all the processed multilingual data consisting of sentence aligned multiple language pairs at once, During inference, we also need to add the aforementioned additional token to each input source sentence of the source data to specify the desired target language.

[1] https://github.com/anoopkunchukuttan/indic_nlp_library/

5 Experiments

In the next sub-sections we describe the experiments we carried out for translating from Hindi to Marathi and from Marathi to Hindi for WIPRO-RIT's WMT 2020 SLT shared task submission.

5.1 Experiment Setup

To handle out-of-vocabulary words and to reduce the vocabulary size, instead of considering words, we consider subword units (Sennrich et al., 2016) by using byte-pair encoding (BPE). In the preprocessing step, instead of learning an explicit mapping between BPEs in the English (EN), Hindi (HI) and Marathi (MR), we define BPE tokens by jointly processing all parallel data. Thus, all derive a single BPE vocabulary. Since HI and MR belong to the similar languages, they naturally share a good fraction of BPE tokens, which reduces the vocabulary size.

We report evaluation results (evaluated by the shared task organizers) of our approach with the released Test data. BLEU (Papineni et al., 2002), RIBES (Isozaki et al., 2010) and TER (Snover et al., 2006) are used to evaluate the performance of all participating systems in the shared task.

Parallel Data	#sentences	C1	C2	P
Filtered SLT	33,923	✓	✓	✓
Filtered EN–HI	961,036	✓	✓	✓
BT EN–HI	5 million	✓	✓	✓
BT HI–MR	5 million		✓	✓

Table 4: The training criteria data statistics of our submitted systems (C1 = Contrastive 1, C2 = Contrastive 2, P = Primary, and BT = Back-translated data).

5.2 Hyper-parameter Setup

We follow a similar hyper-parameter setup for all reported systems. All encoders, and the decoder, are composed of a stack of $N_X = 6$ identical layers followed by layer normalization. Each layer again consists of two sub-layers and a residual connection (He et al., 2016) around each of the two sub-layers. We apply dropout (Srivastava et al., 2014) to the output of each sub-layer, before it is added to the sub-layer input and normalized. Furthermore, dropout is applied to the sums of the word embeddings and the corresponding positional encodings in both encoders as well as the decoder stacks.

We set all dropout values in the network to 0.1. During training, we employ label smoothing with value $\epsilon_{ls} = 0.1$. The output dimension produced by all sub-layers and embedding layers is $d_{model} = 512$. Each encoder and decoder layer contains a fully connected feed-forward network (FFN) having dimensionality of $d_{model} = 512$ for the input and output and dimensionality of $d_{ff} = 2048$ for the inner layers. For the scaled dot-product attention, the input consists of queries and keys of dimension d_k, and values of dimension d_v. As multi-head attention parameters, we employ $h = 8$ for parallel attention layers, or heads. For each of these we use a dimensionality of $d_k = d_v = d_{model}/h = 64$. For optimization, we use the Adam optimizer (Kingma and Ba, 2015) with $\beta_1 = 0.9$, $\beta_2 = 0.98$ and $\epsilon = 10^{-9}$.

The learning rate is varied throughout the training process, and increasing for the first training steps $warmup_{steps} = 16000$ and afterwards decreasing as described in (Vaswani et al., 2017). All remaining hyper-parameters are set analogously to those of the transformer's *base* model. At training time, the batch size is set to 25K tokens, with a maximum sentence length of 256 subwords, and a vocabulary size of 32K. After each epoch, the training data is shuffled. During decoding, we perform beam search with a beam size of 4. We use 32K BPE operations to train our BPE models. We use shared embeddings in all our experiments.

6 Results

We present the results obtained by our systems for Hindi–Marathi in Table 5 and for Marathi–Hindi in Table 6 in terms of BLEU, RIBES, and TER. We apply our proposed method to train multilingual models in three different configurations. Table 4 shows different training data used to train our CONTRASTIVE 1 (C1), CONTRASTIVE 2 (C2) and Primary (P) submissions.

System	BLEU ↑	RIBES ↑	TER ↓
P	16.62	62.45	72.23
C2	15.42	61.02	73.59
C1	13.25	58.51	76.17

Table 5: Results for Hindi to Marathi translation ranked by BLEU score.

System	BLEU ↑	RIBES ↑	TER ↓
P	24.53	66.23	66.39
C2	22.93	65.89	68.11
C1	22.69	65.01	68.13

Table 6: Results for Marathi to Hindi Translation ranked by BLEU score.

CONTRASTIVE 1 (C1) Our CONTRASTIVE 1 submission is a multilingual single system and does not use any monolingual back translation data. The system is trained on the released HI-MR and MR-HI parallel data. In addition to we also use EN-HI parallel data.

CONTRASTIVE 2 (C2) This submission is similar to CONTRASTIVE 1, however in this case we used 5M back-translated Marathi–Hindi and 5M back-translated Hindi–Marathi corpus. Source back-translated sentences begin with an additional token indicating the target language.

PRIMARY (P) Our primary submission is trained using the same setting as we described in CONTRASTIVE 2 system. The difference is our primary system is an ensemble of three different CONTRASTIVE 2 systems initiated with three different random seeds.

7 Conclusion and Future Work

This paper presented the WIPRO–RIT system submitted to the Similar Language Translation shared task at WMT 2020. We presented the results obtained by our system in translating from Hindi to Marathi and Marathi to Hindi. Our primary system achieved competitive performance ranking first in Marathi to Hindi and second in Hindi to Marathi among 22 teams in terms of BLEU score.

In future work, we would like to further explore the similarity between these two languages in translating to other Indo-Aryan languages (e.g. Bengali, Bhojpuri, and Nepali). We expect the models presented in this paper to perform well for other Indo-Aryan language provided that suitable training data is available. Furthermore, we would like to apply and evaluate our method on the two other groups of languages in the WMT SLT 2020 shared task, Romance languages: Catalan, Portuguese, and Spanish, and South Slavic languages: Croatian, Serbian, and Slovene. Finally, we will be incorporating the translation models presented in this paper to CATaLog, an open-source online CAT tool that provides users with both MT and TM outputs (Nayek et al., 2015; Pal et al., 2016).

Acknowledgments

We would like to thank the WMT 2020 SLT shared task organizers for making the Hindi - Marathi data available. We further thank the anonymous WMT reviewers for their insightful feedback and suggestions.

References

Loïc Barrault, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. 2019. Findings of the 2019 Conference on Machine Translation (WMT19). In *Proceedings of WMT*.

Pushpak Bhattacharyya. 2010. IndoWordNet. In *Proceedings of LREC*.

Ondřej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, et al. 2014. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of WMT*.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, et al. 2017. Findings of the 2017 Conference on Machine Translation (WMT17). In *Proceedings of WMT*.

Ondrej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, et al. 2016. Findings of the 2016 Conference on Machine Translation. In *Proceedings of WMT*.

Marta R. Costa-jussà. 2017. Why Catalan-Spanish Neural Machine Translation? Analysis, Comparison and Combination with Standard Rule and Phrase-based Technologies. In *Proceedings of VarDial*.

Marta R. Costa-jussà, Marcos Zampieri, and Santanu Pal. 2018. A Neural Approach to Language Variety Translation. In *Proceedings of VarDial*.

Mihaela Găman, Dirk Hovy, Radu Tudor Ionescu, Heidi Jauhiainen, Tommi Jauhiainen, Krister Lindén, Nikola Ljubešić, Niko Partanen, Christoph Purschke, Yves Scherrer, and Marcos Zampieri. 2020. A Report on the VarDial Evaluation Campaign 2020. In *Proceedings of VarDial*.

Barry Haddow and Faheem Kirefu. 2020. Pmindia - a collection of parallel corpora of languages of india. *arXiv preprint arXiv:2001.09907*.

Salima Harrat, Karima Meftouh, and Kamel Smaili. 2019. Machine Translation for Arabic Dialects (Survey). *Information Processing & Management*, 56(2):262–273.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. *Proceedings of CVPR*.

Hideki Isozaki, Tsutomu Hirao, Kevin Duh, Katsuhito Sudoh, and Hajime Tsukada. 2010. Automatic evaluation of translation quality for distant language pairs. In *Proceedings of EMNLP*.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Diederik P Kingma and Jimmy Lei Ba. 2015. Adam: A Method for Stochastic Optimization. *Proceedings of ICLR*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of ACL*.

Anoop Kunchukuttan. 2020a. Indowordnet parallel corpus. https://github.com/anoopkunchukuttan/indowordnet_parallel.

Anoop Kunchukuttan. 2020b. The IndicNLP Library. https://github.com/anoopkunchukuttan/indic_nlp_library/blob/master/docs/indicnlp.pdf.

Anoop Kunchukuttan, Pratik Mehta, and Pushpak Bhattacharyya. 2018. The IIT Bombay English-Hindi parallel corpus. In *Proceedings of LREC*.

Surafel M Lakew, Aliia Erofeeva, and Marcello Federico. 2018. Neural Machine Translation into Language Varieties. *arXiv preprint arXiv:1811.01064*.

Thazin Myint Oo, Ye Kyaw Thu, and Khin Mar Soe. 2019. Neural machine translation between Myanmar (Burmese) and rakhine (arakanese). In *Proceedings of VarDial*.

Tapas Nayek, Sudip Kumar Naskar, Santanu Pal, Marcos Zampieri, Mihaela Vela, and Josef van Genabith. 2015. CATaLog: New approaches to TM and post editing interfaces. In *Proceedings of NLP4TM*.

Santanu Pal, Sudip Naskar, and Josef van Genabith. 2015. UdS-sant: English–German hybrid machine translation system. In *Proceedings of WMT*.

Santanu Pal, Marcos Zampieri, Sudip Kumar Naskar, Tapas Nayak, Mihaela Vela, and Josef van Genabith. 2016. CATaLog online: Porting a post-editing tool to the web. In *Proceedings of LREC*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of ACL*.

Maja Popović, Alberto Poncelas, Marija Brkic, and Andy Way. 2020. Neural Machine Translation for Translating into Croatian and Serbian. In *Proceedings of the Seventh Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of ACL*.

Pamela Shapiro and Kevin Duh. 2019. Comparing pipelined and integrated approaches to dialectal Arabic neural machine translation. In *Proceedings of VarDial*.

Shashank Siripragada, Jerin Philip, Vinay P. Namboodiri, and C V Jawahar. 2020. A multilingual parallel corpora collection effort for Indian languages. In *Proceedings of LREC*.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A Study of Translation Edit Rate with Targeted Human Annotation. In *Proceedings of AMTA*.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. In *Proceedings of NIPS*.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Ahmed Ali, Suwon Shon, James Glass, Yves Scherrer, Tanja Samardžić, Nikola Ljubešić, Jörg Tiedemann, Chris van der Lee, Stefan Grondelaers, Nelleke Oostdijk, Dirk Speelman, Antal van den Bosch, Ritesh Kumar, Bornini Lahiri, and Mayank Jain. 2018. Language Identification and Morphosyntactic Tagging: The Second VarDial Evaluation Campaign. In *Proceedings of VarDial*.

Marcos Zampieri, Shervin Malmasi, Yves Scherrer, Tanja Samardžić, Francis Tyers, Miikka Silfverberg, Natalia Klyueva, Tung-Le Pan, Chu-Ren Huang, Radu Tudor Ionescu, Andrei Butnaru, and Tommi Jauhiainen. 2019. A Report on the Third VarDial Evaluation Campaign. In *Proceedings of VarDial*.

Neural Machine Translation between similar South-Slavic languages

Maja Popović, Alberto Poncelas
ADAPT Centre, School of Computing
Dublin City University, Ireland
name.surname@adaptcentre.ie

Abstract

This paper describes the ADAPT-DCU machine translation systems built for the WMT 2020 shared task on Similar Language Translation. We explored several set-ups for NMT for Croatian–Slovenian and Serbian–Slovenian language pairs in both translation directions. Our experiments focus on different amounts and types of training data: we first apply basic filtering on the *OpenSubtitles* training corpora, then we perform additional cleaning of remaining misaligned segments based on character n-gram matching. Finally, we make use of additional monolingual data by creating synthetic parallel data through back-translation. Automatic evaluation shows that multilingual systems with joint Serbian and Croatian data are better than bilingual, as well as that character-based cleaning leads to improved scores while using less data. The results also confirm once more that adding back-translated data further improves the performance, especially when the synthetic data is similar to the desired domain of the development and test set. This, however, might come at a price of prolonged training time, especially for multitarget systems.

1 Introduction

Machine translation (MT) between closely related languages is, in principle, less challenging than translation between distantly related languages, but it is still far from being solved. While MT between closely related South-Western Slavic languages, Croatian, Slovenian and Serbian based on the rule-based (RBMT) and the phrase-based (PB-SMT) approaches has been investigated in the last years (Etchegoyhen et al., 2014; Petkovski et al., 2014; Klubička et al., 2016; Arčan et al., 2016; Popović et al., 2016a), to the best of our knowledge, the new state-of-the-art neural machine translation (NMT) has not been investigated yet for these languages.

In this work, we first compare bilingual and multilingual systems in order to determine whether joining Serbian and Croatian data is useful. Afterwards, we investigate additional cleaning of remaining misaligned segments by using character n-gram matching scores (Popović, 2015). The beauty of the method for similar languages is that it can be applied directly to the given training corpus providing matching scores for each pair of the source-target segments. For distant languages, translation of one side of the training corpus would be required. Finally, we make use of monolingual data in each of the three languages by creating additional synthetic parallel training sets via back-translation (Sennrich et al., 2016a; Poncelas et al., 2018; Burlot and Yvon, 2018).

2 Language properties

Common properties All three languages, Croatian, Serbian and Slovenian, belong to the South-Western Slavic branch. As Slavic languages, they have a very rich inflectional morphology for all word classes: six cases and three genders for all nouns, pronouns, adjectives and determiners. For verbs, person and many tenses are expressed by the suffix so that the subject pronoun is often omitted. There are two verb aspects, so that many verbs have perfective and imperfective form(s) depending on the duration of the described action. As for syntax, all three languages have quite a free word order, and neither language uses articles, either definite or indefinite. In addition to this, multiple negation is always used.

Croatian and Serbian Croatian and Serbian exhibit a large overlap in vocabulary and a strong morpho-syntactic similarity so that the speakers can understand each other without difficulties. Nev-

Proceedings of the 5th Conference on Machine Translation (WMT), pages 430–436

ertheless, there is a number of small but notable and also frequently occurring differences between them. The largest differences between the two languages are in vocabulary: some words are completely different, some however differ only by one or two letters. Apart from lexical differences, there are also structural differences mainly concerning verbs: modal verb constructions, future tense, as well as conditional.

Slovenian Even though Slovenian is very closely related to Croatian and Serbian, and the languages share a large degree of mutual intelligibility, a number of Croatian/Serbian speakers may have difficulties with Slovenian and the other way round. The nature of the lexical differences is similar to the one between Croatian and Serbian, namely a number of words is completely different and a number only differs by one or two letters. However, the amount of different words is much larger. In addition to that, the set of overlapping words includes a number of false friends (e.g. *brati* means *to pluck* in Croatian and Serbian but *to read* in Slovenian).

The amount of grammatical differences is also larger and includes local word order, verb mood and/or tense formation, question structure, usage of cases, structural properties for certain conjunctions, as well as some other structural differences. Another important difference is the Slovenian dual grammatical number which refers to two entities (apart from singular for one and plural for more than two). It requires additional set of pronouns, as well as additional sets for noun, adjective and verb inflexion rules not existing either in Croatian or in Serbian.

3 Data

For training, we used publicly available OPUS[1] parallel corpora (Tiedemann, 2012) indicated by the workshop organisers. *OpenSubtitles* is indicated for all translation directions. For Croatian–Slovenian, other corpora are indicated too, but they are either not sentence-aligned (*JW300*) or are extremely noisy (*DGT, MultiParaCrawl*). Therefore, we decided to use only *OpenSubtitles* for all translation directions.

It is worth noting that the organisers also indicated the *SETIMES News* parallel Croatian–Serbian corpus. Developing an additional Croatian–Serbian MT system for converting Serbian data into Croatian and vice versa was shown to be helpful for the PBSMT approach (Popović and Ljubešić, 2014; Popović et al., 2016b). However, our preliminary experiments in this direction indicated that this technique is not helpful for the NMT approach.

[1] http://opus.nlpl.eu/

lang.	set	domain	# sentences
sl-hr	train	*Subtitles*	11 213 386
	dev	*PR publications*	2457
	test	*PR publications*	2582
sl-sr	train	*Subtitles*	11 780 062
	dev	*PR publications*	1259
	test	*PR publications*	1260

Table 1: Corpus statistics.

The original parallel data were filtered in order to eliminate noisy parts: too long segments (more than 100 words), segment pairs with disproportional sentence lengths, segments with more than 1/3 of non-alphanumeric characters, as well as duplicate segment pairs were removed. The statistics of the remaining subtitles together with the development and test sets is shown in Table 1. The development and test sets were provided by the organisers and originate from Public Relations publications of a business intelligence company.

3.1 Additional cleaning of *OpenSubtitles*

While a large number of noisy parts and misaligned segments was removed from *OpenSubtitles* by the basic filtering procedure, a number of misaligned segments still remained. In order to remove these, we applied additional cleaning based on the character n-gram F-score chrF usually used for MT evaluation (Popović, 2015). For the purpose of cleaning, the chrF score is calculated for each pair of segments in the training data. Due to similarity between the languages, the scores between the properly aligned segments are higher than the scores of misaligned segments. Nevertheless, the languages are sufficiently different so that some properly aligned short segments (or single words) can have low scores, too. Still, if those words also appear in longer sentences, they will not be removed. Preliminary experiments with different thresholds showed that keeping the segments with the chrF score equal or greater than 20 is the best option.

3.2 Using monolingual data

In addition to the parallel *OpenSubtitles* corpora, we also used the monolingual data in each of the three languages which were indicated by the organisers, namely the mixed-domain data collected from Web, *hrWac*, *slWac* and *hrWac* (Ljubešić and Erjavec, 2011; Ljubešić and Klubička, 2014). As a first step, we removed too long and too short sentences, keeping those between 5 and 60 words. Then, we removed sentences with more than 1/3 of non-alphanumeric characters, sentences with URLs, as well as duplicate sentences.

Then, we wanted to rank these sentences according to the relevance for our experiments, namely according to their similarity to the development corpus. For this purpose, we used Feature Decay Algorithm (FDA) (Biçici and Yuret, 2011). This method iteratively selects sentences from an initial set S based on the number of n-grams which overlap with an in-domain text $Seed$ and adds these sentences to a selected set Sel. In addition, in order to promote a diversity, after a sentence is selected, its n-grams suffer a penalisation so that they are less likely to be selected in the following iterations. The default FDA system halves the score of an n-gram each time it is selected. Therefore the score of a sentence s is computed as in Equation (1):

$$score(s, Seed, Sel) = \frac{\sum_{ngr \in \{s \bigcap Seed\}} 0.5^{C_{Sel}(ngr)}}{\text{length(s)}} \quad (1)$$

where Sel is the set of sentences that have been selected and $C_Sel(ngr)$ is the count of occurrences of the *n*-gram ngr. At the end, the set S is converted into the set Sel containing the same sentences, but ranked according to their relevance.

For our experiments, the *hrWac*, *slWac* and *srWac* corpora represented the sets S, and the development sets in the corresponding target language were used as $Seed$.

Back-translated synthetic parallel corpora After ranking the monolingual corpora by FDA, back-translation was applied in order to create additional parallel training corpora. For each translation direction, the first two million best ranked sentences in the target language were translated into the source language by the corresponding NMT system.

Translation from Slovenian: The first two million best ranked Serbian sentences and the first two million best ranked Croatian sentences were translated into Slovenian.

Translation into Slovenian: Slovenian is the target language for two translation directions, and we wanted to have equally relevant Slovenian sentences for both directions. Therefore, we did not take the first two million sentences for one source language and the second two million for the other, because the Slovenian sentences for the first source language would be more relevant than those for the second source language. Instead, we took the first four million best ranked Slovenian sentences, and then translated every odd sentence into Serbian and every even sentence into Croatian.

4 MT systems

All our systems are built using the Sockeye implementation (Hieber et al., 2018) of the Transformer architecture (Vaswani et al., 2017). The systems operate on sub-word units generated by byte-pair encoding (BPE) (Sennrich et al., 2016b). We set the number of BPE merging operations at 32000. We use shared vocabularies between the languages because they are similar. Multilingual systems are built using the same technique as (Johnson et al., 2017) and (Aharoni et al., 2019), namely adding a target language label "SR" or "HR" to each source sentence. We investigated the following set-ups:

1. Systems trained on *OpenSubtitles*

   The four bilingual systems, HR→SL, SR→SL, SL→HR and SL→SR, are trained separately for each language pair and each translation direction on about 11M parallel segments.

   The multisource system HR+SR→SL is trained for translation into Slovenian by joining Serbian and Croatian sources and removing duplicates, thus resulting in 20.2M parallel segments.

   The multitarget system SL→HR+SR is trained for translation from Slovenian on the reversed corpus of 20.2M segments with target language identificators "SR" and "HR" added to the source side.

2. Systems trained on cleaned *OpenSubtitles*

   Two multilingual systems HR+SR→SL_CLEAN and SL→HR+SL_CLEAN are trained on joint *OpenSubtitles* corpora additionally cleaned by the chrF score. The

cleaned corpus consists of 10.8M segments (instead of 20.2M).

3. Systems trained on cleaned *OpenSubtitles* and synthetic back-translated parallel *Wac* data

   Two multilingual systems HR+SR→SL_CLEAN+BT and SL→HR+SR_CLEAN+BT are trained on joint cleaned *OpenSubtitles* corpora together with the corresponding synthetic back-translated data selected from *hrWac*, *slWac* and *srWac*. The monolingual data was back-translated by the corresponding systems trained on cleaned *OpenSubtitles*. The training corpora consist of 14.8M segments.

5 Results

We evaluate our systems using the following three automatic overall evaluation scores: sacreBLEU (Post, 2018), chrF (Popović, 2015) and characTER (Wang et al., 2016). The BLEU score is used because of the long tradition. The two character level scores are shown to correlate much better with human assessments (Bojar et al., 2017; Ma et al., 2018), especially for morphologically rich languages. In addition, the chrF score is recommended as a replacement for BLEU in a recent detailed study encompassing a number of automatic MT metrics (Mathur et al., 2020). In addition to the automatic MT evaluation scores, for each of the systems we report the size of the training corpus and the training time.

Table 2 shows the results both on the development and on the test set for each of the four translation directions. First of all, it can be seen that the automatic scores are relatively low given the similarity of the languages. One reason is domain/genre discrepance between the training and the development/test sets. Another possible reason is the nature of the *OpenSubtitles* corpus. The majority of non-English texts in *OpenSubtitles* are namely human translations from English originals. Therefore, for translation from English, the source language is the original one and the target language is its human translation.[2] On the other hand, for translation not involving English, both sides are human translations, which can have a strong impact on performance (Kurokawa et al., 2009; Vyas et al., 2018; Zhang and Toral, 2019). These effects should be investigated in future work.

[2] And other way round for translation into English.

Results on the development set For the systems trained on *OpenSubtitles*, it can be seen that for each translation direction, multilingual systems yield better automatic scores than bilingual systems at the cost of slightly prolonged training time (from about 3 days to 3-4 days). Therefore we choose the two multilingual systems HR+SR→SL and SL→HR+SR as the baselines and we did not keep the bilingual systems for further experiments.

The chrF cleaning of *OpenSubtitles* reduces the size of the corpus and the training time while slightly improving automatic scores. The reduction in time is slightly smaller for the multitarget translation from Slovenian (down to 2-3 days) than for the multisource translation into Slovenian (down to less than 2 days).

Adding the back-translated data from *Wac* improves the automatic scores for more than 10 points for multisource translation (into Slovenian) and for 5 to 10 points for multitarget translation (from Slovenian). This could be expected, especially since the monolingual data was chosen to be similar to the development data. Nevertheless, this large improvement comes at a price. Although the increase of the corpus is not very large, from 10.8M to 14.8M, the training time increases to (more than) 3 days. It can be noted that for some set-ups, the multitarget system needs more training time. The probable reason is the diversity of the target part of the training corpus – the system has to deal with two target languages, and when synthetic data is added, also with two different domains/genres for each of them.

Results on the test set Based on the results on the development set, we submitted the outputs of the systems with back-translated data (HR+SR→SL_CLEAN+BT, SL→HR+SR_CLEAN+BT) as primary submissions. The outputs of the systems trained on cleaned data (HR+SR→SL_CLEAN, SL→HR+SR_CLEAN) were submitted as first contrastive, and the outputs of the baseline multilingual systems (HR+SR→SL, SL→HR+SR) as second contrastive submissions. The test sets were not at all translated by the initial bilingual systems, therefore the results are not available.

It can be seen that the tendencies for the test set are almost the same as for the development set. The only difference is the larger improvement obtained by cleaning *OpenSubtitles* with the chrF scores. Further detailed analysis involving manual inspec-

(a) Croatian→Slovenian

training			dev, hr→sl			test, hr→sl		
system	size	time	BLEU	chrF	chrTER	BLEU	chrF	chrTER
HR→SL	11.2M	∼3 days	38.5	65.7	29.4	/	/	/
HR+SR→SL	20.2M	3-4 days	38.8	65.9	29.5	34.7	62.2	34.5
HR+SR→SL_CLEAN	10.8M	<2 days	39.7	66.5	27.0	*37.1*	*65.2*	*28.2*
HR+SR→SL_CLEAN+BT	14.8M	∼3 days	**53.9**	**77.7**	**18.9**	**51.9**	**76.4**	**20.0**

(b) Serbian→Slovenian

training			dev, sr→sl			test, sr→sl		
system	size	time	BLEU	chrF	chrTER	BLEU	chrF	chrTER
SR→SL	11.8M	∼3 days	40.6	67.2	30.3	/	/	/
HR+SR→SL	20.2M	3-4 days	42.1	68.3	28.5	37.7	64.1	33.5
HR+SR→SL_CLEAN	10.8M	<2 days	42.2	68.6	26.9	*41.2*	*68.1*	*26.5*
HR+SR→SL_CLEAN+BT	14.8M	∼3 days	**58.0**	**80.4**	**18.5**	**55.2**	**78.4**	**19.1**

(c) Slovenian→Croatian

training			dev, sl→hr			test, sl→hr		
system	size	time	BLEU	chrF	chrTER	BLEU	chrF	chrTER
SL→HR	11.2M	∼3 days	33.4	62.6	33.0	/	/	/
SL→HR+SR	20.2M	3-4 days	36.0	63.8	32.6	30.3	58.9	40.0
SL→HR+SR_CLEAN	10.8M	2-3 days	36.9	65.2	28.6	*35.7*	*64.4*	*28.8*
SL→HR+SR_CLEAN+BT	14.8M	>3 days	**46.1**	**72.7**	**22.8**	**45.1**	**72.3**	**23.3**

(d) Slovenian→Serbian

training			dev, sl→sr			test, sl→sr		
system	size	time	BLEU	chrF	chrTER	BLEU	chrF	chrTER
SL→SR	11.8M	∼3 days	33.3	62.3	34.3	/	/	/
SL→HR+SR	20.2M	3-4 days	34.8	63.4	33.4	32.0	60.0	36.4
SL→HR+SR_CLEAN	10.8M	2-3 days	35.5	64.2	31.5	*37.0*	*65.1*	*28.2*
SL→HR+SR_CLEAN+BT	14.8M	>3 days	**45.5**	**73.3**	**23.4**	**47.6**	**73.6**	**22.1**

Table 2: Results: Croatian→Slovenian (a), Serbian→Slovenian (b), Slovenian→Croatian (c) and Slovenian→Serbian: corpus size, training time, and the three automatic MT evaluation scores (BLEU, chrF and characTER).

tion is needed to better understand this difference.

6 Summary and outlook

This work investigates different set-ups for training NMT systems for translation between three closely related South-Slavic languages: Slovenian on one side, and Serbian and Croatian on the other side. We explore different sizes and types of training corpora, as well as bilingual and multilingual systems. Our results show that for all translation directions, multilingual systems with joint Croatian and Serbian data perform better than bilingual systems. The results also show that cleaning misaligned segments using character n-gram matching (chrF score) represents a fast and useful method for closely related languages, which improved the evaluation scores while reducing corpus size and training time. Finally, we confirm that adding back-translated synthetic data, which is the usual practice in neural machine translation, can yield large improvements of evaluation scores also for these languages. Nevertheless, for multitarget translation, it might result in a prolonged training time due to increased variety of the target language side.

Future work should include more genres and domains, as well as detailed analysis of errors and problems in order to further improve the performance of NMT between South Slavic languages.

Acknowledgments

The ADAPT SFI Centre for Digital Content Technology (www.adaptcentre.ie) is funded by Science Foundation Ireland through the SFI Research Centres Programme and is co-funded under the European Regional Development Fund (ERDF) through Grant 13/RC/2106.

References

Roee Aharoni, Melvin Johnson, and Orhan Firat. 2019. Massively multilingual neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2019)*, pages 3874–3884, Minneapolis, Minnesota.

Mihael Arčan, Maja Popović, and Paul Buitelaar. 2016. Asistent – A Machine Translation System for Slovene, Serbian and Croatian. In *Proceedings of the Tenth Conference on Language Technologies and Digital Humanities (JDTH 2016)*, pages 13–20, Ljubljana, Slovenia.

Ergun Biçici and Deniz Yuret. 2011. Instance selection for machine translation using feature decay algorithms. In *Proceedings of the Sixth Workshop on Statistical Machine Translation (WMT 2011)*, pages 272–283, Edinburgh, Scotland.

Ondřej Bojar, Yvette Graham, and Amir Kamran. 2017. Results of the WMT17 metrics shared task. In *Proceedings of the Second Conference on Machine Translation (WMT 2017)*, pages 489–513, Copenhagen, Denmark.

Franck Burlot and François Yvon. 2018. Using Monolingual Data in Neural Machine Translation: a Systematic Study. In *Proceedings of the 3rd Conference on Machine Translation (WMT 2018)*, pages 144–155, Belgium, Brussels.

Thierry Etchegoyhen, Lindsay Bywood, Mark Fishel, Panayota Georgakopoulou, Jie Jiang, Gerard van Loenhout, Arantza del Pozo, Mirjam Sepesy Maučec, Anja Turner, and Martin Volk. 2014. Machine translation for subtitling: A large-scale evaluation. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC 2014)*, pages 46–53, Reykjavik, Iceland.

Felix Hieber, Tobias Domhan, Michael Denkowski, David Vilar, Artem Sokolov, Ann Clifton, and Matt Post. 2018. The sockeye neural machine translation toolkit at AMTA 2018. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Papers)*, pages 200–207, Boston, MA. Association for Machine Translation in the Americas.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Filip Klubička, Gema Ramírez-Sánchez, and Nikola Ljubešić. 2016. Collaborative development of a rule-based machine translator between Croatian and Serbian. In *Proceedings of the 19th Annual Conference of the European Association for Machine Translation (EAMT 2016)*, pages 361–367.

David Kurokawa, Cyril Goutte, and Pierre Isabelle. 2009. Automatic detection of translated text and its impact on machine translation. In *In Proceedings of MT Summit XII*, pages 81–88, Ottawa, Canada.

Nikola Ljubešić and Tomaž Erjavec. 2011. hrWaC and slWac: Compiling Web Corpora for Croatian and Slovene. In *Proceedings of the 14 Conference on Text, Speech and Dialogue (TSD 2011)*, Lecture Notes in Computer Science, pages 395–402, Pilsen, Czech Republic. Springer.

Nikola Ljubešić and Filip Klubička. 2014. {bs,hr,sr}WaC – web corpora of Bosnian, Croatian and Serbian. In *Proceedings of the 9th Web as Corpus Workshop (WaC-9)*, pages 29–35, Gothenburg, Sweden.

Qingsong Ma, Ondřej Bojar, and Yvette Graham. 2018. Results of the WMT18 metrics shared task: Both characters and embeddings achieve good performance. In *Proceedings of the Third Conference on Machine Translation (WMT 2018)*, pages 671–688, Belgium, Brussels.

Nitika Mathur, Timothy Baldwin, and Trevor Cohn. 2020. Tangled up in BLEU: Reevaluating the evaluation of automatic machine translation evaluation metrics. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020)*, pages 4984–4997, Online.

Filip Petkovski, Francis Tyers, and Hrvoje Peradin. 2014. Shallow-transfer rule-based machine translation for the western group of south slavic languages. In *Proceedings of the 9th Workshop on Free/open-Source Language Resources for the Machine Translation of Less-Resourced Languages (SaLTMil 2014)*, pages 25–30, Reykjavik, Iceland.

Alberto Poncelas, Dimitar Shterionov, Andy Way, Gideon Maillette de Buy Wenniger, and Peyman Passban. 2018. Investigating Back translation in Neural Machine Translation. In *Proceedings of the 21st Annual Conference of the European Association for Machine Translation (EAMT 2018)*, Alicante, Spain.

Maja Popović. 2015. chrF: character n-gram f-score for automatic MT evaluation. In *Proceedings of the*

Tenth Workshop on Statistical Machine Translation, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Maja Popović, Mihael Arčan, and Filip Klubička. 2016a. Language related issues for machine translation between closely related south Slavic languages. In *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial3)*, pages 43–52, Osaka, Japan.

Maja Popović, Kostadin Cholakov, Valia Kordoni, and Nikola Ljubešić. 2016b. Enlarging scarce in-domain English-Croatian corpus for SMT of MOOCs using Serbian. In *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial3)*, pages 97–105, Osaka, Japan.

Maja Popović and Nikola Ljubešić. 2014. Exploring cross-language statistical machine translation for closely related south Slavic languages. In *Proceedings of the EMNLP'2014 Workshop on Language Technology for Closely Related Languages and Language Variants (LT4CloseLang 2014)*, pages 76–84, Doha, Qatar.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving Neural Machine Translation Models with Monolingual Data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*, pages 86–96, Berlin, Germany.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*, pages 1715–1725, Berlin, Germany.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC 2012)*, pages 2214–2218, Istanbul, Turkey.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st Conference on Neural Information Processing Systems (NeurIPS 2017)*, pages 5998–6008, Long Beach, CA.

Yogarshi Vyas, Xing Niu, and Marine Carpuat. 2018. Identifying semantic divergences in parallel text without annotations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2018)*, pages 1503–1515, New Orleans, Louisiana.

Weiyue Wang, Jan-Thorsten Peter, Hendrik Rosendahl, and Hermann Ney. 2016. CharacTer: Translation Edit Rate on Character Level. In *Proceedings of the 1st Conference on Machine Translation (WMT 2016)*, pages 505–510, Berlin, Germany.

Mike Zhang and Antonio Toral. 2019. The effect of translationese in machine translation test sets. In *Proceedings of the Fourth Conference on Machine Translation (WMT 2019)*, pages 73–81, Florence, Italy. Association for Computational Linguistics.

Infosys Machine Translation System for WMT20 Similar Language Translation Task

Kamalkumar Rathinasamy, Amanpreet Singh, Balaguru Sivasambagupta, Prajna Prasad Neerchal, Vani Sivasankaran
Infosys Limited
{kamalkumar_r,amanpreet.singh04,balaguru.s,prajna.neerchal,vani.s}@infosys.com

Abstract

This paper describes Infosys' submission to the WMT20 Similar Language Translation shared task. We participated in Indo-Aryan language pair in the language direction Hindi to Marathi. Our baseline system is byte-pair encoding based transformer model trained with the fairseq sequence modeling toolkit. Our final system is an ensemble of two transformer models, which ranked first in the WMT20 evaluation. One model is designed to learn the nuances of translation of this low resource language pair by taking advantage of the fact that the source and target languages are the same alphabet languages. The other model is the result of experimentation with the proportion of back-translated data to the parallel data to improve translation fluency.

1 Introduction

Neural Machine Translation (Bahdanau et al., 2015; Vaswani et al., 2017) is the most popular approach for machine translation. Transformer-based NMT has outperformed many recurrent neural network based models. There is scope for improvement in NMT, particularly for low-resource language pairs.

Our techniques are experimented on the fairseq sequence modeling toolkit (Ott et al., 2019) for NMT. Our system is an ensemble of two transformer-based models. One designed for low-resource language pairs by taking advantage that both are same alphabet languages. The other model is built after experimenting on renowned back-translation technique (Sennrich et al., 2016a) by exploiting target monolingual data.

2 Data

Hindi-Marathi bitext data contains ∼49K sentence pairs. Target monolingual data comprises of 326K Newscrawl sentences and 10,839K raw sentences.

2.1 Data Preprocessing

Typical training sentence pairs comprises of a source and a target sentence. There are ∼1K training sentence pairs where source or target contains multiple sentences delimited by '/'. Matching pair for these sentences is derived based on the proximity of token lengths between source and target sentence.

Non-printable characters are removed, punctuations are normalized, and the data is tokenized, with the Moses tokenizer. Byte-pair encoding (BPE) has been adopted (Sennrich et al., 2016a) to build source and target sub-word vocabularies of size 22.5K and 32.8K respectively, when configured to construct with 60K symbols.

2.2 Data filtering

2.2.1 Bitext data

Sentences with more than 175 words, sentences with no words, and sentence pairs exceeding length ratio of 1.5 are removed from training data. This eliminated around 18% of the overall real bitext data.

2.2.2 Synthetic data

CommonCrawl n-grams raw monolingual files are processed[1] to remove sentences with invalid characters, strip leading and trailing whitespaces, and remove duplicate sentences.

3 System Overview

Our Hindi-Marathi primary system is an ensemble of two transformer models. One is back-translated model and the other model is trained on anonymized data.

[1] https://github.com/kpu/preprocess

Proceedings of the 5th Conference on Machine Translation (WMT), pages 437–441
Online, November 19–20, 2020.

3.1 Base Model Architecture and Hyperparameters

Our model is built using fairseq[2] (Ott et al., 2019) toolkit. The Transformer, an encoder-decoder architecture (Vaswani et al., 2017), with 6 layers for the encoder and 6 layers for the decoder, and with 8 heads in all multi-head attention layers, is our base model. Embedding dimension is set to 512 and feed-forward size (FFN) is set to 2048. Our model is trained on single GPU with maximum tokens per GPU set to 4096. The batch size multiplier is set to 8. Dropout probability of 0.3 and label smoothing probability of 0.1 is applied to avoid overfitting. Adam optimizer is used with $\beta_1 = 0.9$, $\beta_2 = 0.98$. The model is trained with an initial learning rate of 5e-4 and 4000 warm-up updates. The ensemble model prepared by averaging last 3 checkpoints is used for inferencing. Reported detokenized test BLEU is 9.13 for the provided dev dataset.

Parameters are tuned, following Baquero-Arnal et al. (2019). Threshold frequency is set such that only tokens occurring at least 10 times in the training data will be part of the vocabulary. Maximum tokens per GPU is set to 4000 and the batch size multiplier is fixated to 4 to set an effective batch size of 16000 tokens with dropout probability of 0.1. This led to improved performance. Reported detokenized test BLEU is 14.13 and hence these settings are adapted.

3.2 Backtranslation

Back-translation is a popularly adapted data augmentation technique which aids in building better NMT systems, especially for low resource language pairs by leveraging monolingual corpora (Sennrich et al., 2016a). An intermediate system is first trained on parallel data which is used to translate target monolingual data into source language. Sampling is used as a method for inference (Edunov et al., 2018). Synthetic parallel data is constructed from the intermediate system generated synthetic source while the target is the provided monolingual data. The Bitext data filters are also applied to synthetic data but only removed sentences with more than 250 words. New training data is constructed by appending this synthetic parallel data to real bitext data and a final system that will translate from the source to the target language will be trained.

[2]https://github.com/pytorch/fairseq

3.2.1 Bitext and Synthetic corpora proportion

Related Work Real to synthetic parallel data close to 1-to-1 proportion works best for Sennrich et al. (2016a). Junczys-Dowmunt et al. (2016), also chose 1-to-1 ratio of real to synthetic parallel data for English-Russian news translation task. It is also known from past experiments that increasing the ratio of synthetic training data erratically, degrades system performance, depending on quality and domain of synthetic data (Sennrich et al., 2016a; Currey et al., 2017; Poncelas et al., 2018).

In contrast, experiments conducted by Stahlberg et al. (2018), shows that performance of system does not reduce as long as the ratio of real parallel to synthetic parallel data does not exceed 1-to-8 (1.6M out of 3M Turkish monolingual data is preferred for training along with 0.2M of parallel corpus for English-Turkish). Fadaee and Monz (2018), claims, 1-to-5 real to synthetic parallel data ratio achieved best performance in news translation task for German-English with 4.5M parallel corpus.

This limits from taking advantage of all available monolingual corpus. Only a small portion of it can be used as synthetic parallel training data. Oversampling (Chu et al., 2017; Junczys-Dowmunt and Grundkiewicz, 2018) real parallel data can overcome this problem. By oversampling primary parallel data equivalent to the synthetic parallel data from all monolingual data, effective 1-to-1 ratio of bitext and synthetic parallel data can be retained.

Experiment 1-to-1 ratio of bitext to synthetic data is chosen after experimentation with ratios (see Table 1, Figure 1).

Ratio	BLEU
Baseline (1:0)	14.13
1:0.5	18.08
1:1.0	18.76
1:2.5	16.20
1:5.0	14.49
All monolingual data (1:78.0)	11.01

Table 1: BLEU score for different bitext and synthetic corpora proportion

It is crucial to find the ideal proportion of synthetic data to use. Utilization of all available out-of-domain and raw monolingual corpora to the maximum effect can be further explored.

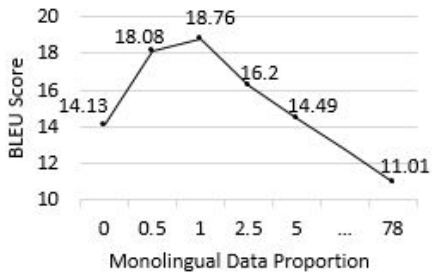


Figure 1: BLEU score for different bitext and synthetic corpora proportion

3.2.2 Out-of-domain data

Handling OOV BPE is applied on monolingual target data using byte pairs learnt during bitext BPE operation. Out-of-vocabulary (OOV) tokens in BPE applied monolingual target data are the tokens not in bitext vocabulary. These out-of-vocabulary tokens are replaced by a special symbol UNK in the monolingual target data (see Table 2).

Considered	Filtered	OOV
25K	19K	16.5%
50K	35K	17.9%
100K	72K	22.0%
300K	172K	21.2%
11.2M	2.5M	23.0%

Table 2: Out-of-domain words in target monolingual data. "Considered" represents the amount of target monolingual data used for study. "Filtered" represents the amount of target monolingual data after applying filters.

Experiment Since the intermediate model spot UNK symbol in the inputs during inferencing, inferenced data also contains UNK symbol.

Gulcehre et al. (2015), claims to eliminate monolingual sentences with more than 10% UNK symbols for better performance. Sennrich et al. (2016b), claims to handle rare/unseen words by representing it in a sequence of sub-word units using existing vocabulary that was learnt on the parallel data. Our systems are experimented by excluding sentences with UNK symbol.

Systems are trained with different proportions of real to synthetic data by eliminating all sentence pairs containing UNK in training data. Table 3 shows the study of model performance before and after removing sentence pairs containing UNK. 1-to-1 proportion of real and synthetic data with out-of-vocabulary tokens masked by UNK symbol scored best (18.76) out of all outcomes.

Ratio	All Data	Data without UNK
1:0.5	18.08	17.73
1:1.0	18.76	18.34
1:5.0	14.49	16.60

Table 3: BLEU scores on models with and without removing sentences containing UNK

3.3 Anonymization

Analysis of the results of the model achieved 18.76 BLEU score, reveals that the translation accuracy is negatively impacted when UNK is generated. This is handled by building another model with bitext data only, where the similarity between source and target languages are anonymized by masking. This approach enables the model to specifically focus on learning the nuances of translation only (i.e.., enables the model to focus on the specific section in the source sentence that gets altered during translation).

Language pair comprising same alphabetic languages contains same words between them carrying similar meaning. Numbers, names, geographic names, etc., also holds same script. i.e. tokens that are not language specific. The approach here is to anonymize those words which are equally present in source and target sentences. One special character is used to mask all those tokens. The special character is chosen in place of a special word to eliminate the possibility of splitting the special word during sub-word tokenization.

This approach reduces the vocabulary size and the learning parameters of the model, preserving the context. This results in transforming sentences which appeared to be different in its raw form into duplicate sentences in its anonymized form, which are then deduplicated.

Hi-Mr track with ∼49K training sentences without masking technique generated source and target vocabulary of size 22.5K and 32.8K respectively. Anonymization reduced source and target vocabulary size to 20.9K and 31.0K respectively. This approach resulted in improvement of BLEU score by 1.2 over baseline. The impact of this approach is proportional to the similarity of source and target languages. The key observation is that this model performed better at translation of sentences that are translated poorly (with UNK tokens) by back-translation model.

3.4 Stacking

Benefits of both the masking systems (masking OOV tokens with UNK symbol and masking similar tokens) are attained through stacking. Model trained on anonymized parallel data and the model trained on real bitext plus synthetic parallel data are ensembled to achieve 19.76 BLEU with Dev data.

3.5 Post-processing

The anonymized words are preserved before inferencing and the inference results are decoded by replacing the special symbols with the preserved anonymized tokens followed by BPE detokenization.

4 Results

Our novel anonymization technique improved BLEU by 1.2. Optimal proportion of back-translated data improved BLEU by 3.5. Ensembling best systems improved BLEU by 1.0. (See Table 4)

System	Dev BLEU
Baseline	9.13
+hyperparameter tuning	14.13
+anonymization	15.46
Baseline	9.13
+hyperparameter tuning	14.13
+backtranslation	18.76
Ensemble	19.71

Table 4: BLEU scores on Hindi-Marathi

Our final submission to the competition in Hindi-Marathi track achieved 18.26 BLEU and ranked first among all submissions.

5 Conclusion

This paper describes the techniques involved in our system submitted for the WMT20 Similar Language Translation task by Infosys. This winning Hindi-Marathi translation system is built based on NMT and evaluated based on the metric, BLEU.

The domain-based data preprocessing and filtering techniques eases model learning. Adopting novel approach of anonymizing language agnostic tokens aided our system to focus more on tokens that matters in the translation. It is highly observed that the ratio of monolingual data used against bitext data plays a vital role in back-translated models. Improving translation accuracy and language fluency by utilizing all available out-of-domain monolingual corpora to the maximum effect can be further explored.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR*, San Diego, USA.

Pau Baquero-Arnal, Javier Iranzo-Sánchez, Jorge Civera, and Alfons Juan. 2019. The mllp-upv spanish-portuguese and portuguese-spanish machine translation systems for wmt19 similar language translation task. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 179–184, Florence, Italy.

Chenhui Chu, Raj Dabre, and Sadao Kurohashi. 2017. An empirical comparison of domain adaptation methods for neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 385–391, Vancouver, Canada.

Anna Currey, Antonio Valerio Miceli-Barone, and Kenneth Heafield. 2017. Copied monolingual data improves low-resource neural machine translation. In *Proceedings of the Second Conference on Machine Translation*, pages 148–156, Copenhagen, Denmark.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. pages 489–500.

Marzieh Fadaee and Christof Monz. 2018. Back-translation sampling by targeting difficult words in neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 436–446, Brussels, Belgium.

Caglar Gulcehre, Orhan Firat, Kelvin Xu, Kyunghyun Cho, Loic Barrault, Huei-Chi Lin, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2015. On using monolingual corpora in neural machine translation. *CoRR*, abs/1503.03535.

Marcin Junczys-Dowmunt, Tomasz Dwojak, and Rico Sennrich. 2016. The amu-uedin submission to the wmt16 news translation task: Attention-based nmt models as feature functions in phrase-based smt. In *Proceedings of the First Conference on Machine Translation (WMT16)*, Berlin, Germany.

Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2018. Ms-uedin submission to the wmt2018 ape shared task: Dual-source transformer for automatic post-editing. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 822–826, Belgium, Brussels.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, USA.

A Poncelas, D Shterionov, A Way, GM de Buy Wenniger, and P Passban. 2018. Investigating back-translation in neural machine translation. arxiv 2018. *CoRR*, abs/1804.06189.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 86–96, Berlin, Germany.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 1715–1725, Berlin, Germany.

Felix Stahlberg, James Cross, and Veselin Stoyanov. 2018. Simple fusion: Return of the language model. pages 204–211.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Document Level NMT of Low-Resource Languages with Backtranslation

Sami Ul Haq[1], **Sadaf Abdul Rauf**[2,3], **Arslan Shoukat**[1] **and Abdullah Saeed**[4]
[1] National University of Sciences and Technology, Pakistan
[2] Fatima Jinnah Women University, Pakistan
[3] LIMSI-CNRS, France
[4] COMSATS University, Pakistan
{sadaf.abdulrauf,abdullahsaeed98}@gmail.com
{sami.ulhaq,arslanshaukat}@ceme.nust.edu.pk

Abstract

This paper describes our system submission to WMT20 shared task on similar language translation. We examined the use of document-level neural machine translation (NMT) systems for low-resource, similar language pair Marathi−Hindi. Our system is an extension of state-of-the-art Transformer architecture with hierarchical attention networks to incorporate contextual information. Since, NMT requires large amount of parallel data which is not available for this task, our approach is focused on utilizing monolingual data with back translation to train our models. Our experiments reveal that document-level NMT can be a reasonable alternative to sentence-level NMT for improving translation quality of low resourced languages even when used with synthetic data.

1 Introduction

With the widespread use of MT systems in commercial and research community, there is an increased attention to train NMT models for direct translation between language pairs other than English Barrault et al. (2019). This is because of the growing need to translate between pairs of similar languages without considering English as pivot language. The task is to overcome the challenge of limited availability of parallel data by exploiting the advantages of similarity between languages when building machine translation models. Similar languages have the advantage of having some magnitude of common information such as lexical and semantic structures. A number of research studies have been published to exploit commonalities when translating text between close language pairs Pourdamghani and Knight (2017); Lakew et al. (2018); Costa-jussà (2017).

This paper describes our system submission at WMT shared Similar Language Translation task[1] which focuses on improving translation quality of similar languages in low-resource setting, the detail of task is provided in Barrault et al. (2019). This year's task includes five pairs of languages from three different language families i.e. Indo-Aryan, Romance and South-Slavic languages; we participated for Hindi-Marathi language pair. Since we are using NMT which requires large bitext, we need to alleviate this specific problem of bitext shortage. Sennrich et al. (2016) introduced an approach to utilize monolingual data using back translation. This requires a machine translation system in opposite direction to generate synthetic parallel corpora from target side monolingual text.

Our work is an attempt to investigate the translation of a similar language pair (Marathi-Hindi) using document-level NMT and back translation. We participated under team name "FJWU_NUST". We submitted one constrained system i.e. we only used the parallel and monolingual data provided by WMT20[2] organizers to train and evaluate our models. We train and evaluate NMT systems in both directions (i.e. HI⇒MR and MR⇒HI) but our submission to similar language shared task comprises of MR⇒HI systems only.

The rest of the paper is structured as follows: In Section 2 we give a brief background of document-level NMT, Section 3 presents utilization of monolingual data, Section 4 and 5 present our experimental setup and results. We conclude the paper in Section 6.

2 Document-Level NMT

Standard NMT works by translating individual sentences and focuses on short context windows while ignoring cross-sentence links and dependencies Xiong et al. (2019). Document-level NMT aims to consider discourse dependencies across sentences

[1] http://www.statmt.org/wmt20/similar.html

[2] http://www.statmt.org/wmt20/.

Proceedings of the 5th Conference on Machine Translation (WMT), pages 442–446

to capture document wide context. Most recently, there has been great interest in modelling larger context in standard NMT (Voita et al., 2018; Wang et al., 2017; Tu et al., 2018; Maruf and Haffari, 2017; Bawden et al., 2017; Jean et al., 2017; Chen et al., 2020). Cache based Tu et al. (2018) memory models can be used to hold rich information, can also provide the context of document during translation. Memory networks keep the representation of a set of words in cache to provide contextual information to NMT in the form of words. Kuang et al. (2017) used two caches, dynamic cache to capture dynamic context by storing words of translated sentence and topic cache which stores topical words of target side from entire document. Through a gating mechanism, the probability of NMT model and cache based neural model is combined to predict the next word. Miculicich et al. (2018) has proposed to use hierarchical attention network (HAN) Yang et al. (2016) to provide dynamic contextual information to NMT during translation. HANs are used on both sides, encoder and decoder to integrate source and target side context in NMT. In contrast to Recurrent Neural Networks (RNN), HANs provide dynamic access to contextual information during training and evaluation.

Similarly, Maruf and Haffari (2018) used pretrained RNN encoder to attach global source and target context to sentence based NMT. Zhang et al. (2018) has shown that integration of short context (2 sentences) outperforms existing cache based RNNSearch model. Voita et al. (2018) introduce a context aware NMT model with additional multi-head attention component, in which they control and analyze the flow of information from the extended context to the translation model.

Stojanovski and Fraser (2020) studied the use of Transformer based document-level models adoptable to novel (zero-resource) domains. They have shown the implicit domain adaptation of document-level NMT models trained on multi-domain data, is capable of capturing large context. The challenge of translating single sentences efficiently while keeping models insensitive to enlarge and noisy context is addressed by Zheng et al. (2020). To make general purpose context-aware MT, both for short and long sentences, they opt for having independent global and local context integration into sentence based NMT.

3 Utilizing Monolingual Data

Large amounts of monolingual resources are generally available for a multitude of languages. Back translation is considered a well known approach to mitigate the need of large parallel corpora by automatically translating target language monolingual data to source language Sennrich et al. (2016). Back translation requires a MT system in opposite direction, where target side monolingual data is translated into source text to generate synthetic parallel training data. Several techniques exists to utilize monolingual text for improving NMT (Abdul-Rauf et al., 2016; Zhang and Zong, 2016; Currey et al., 2017; Domhan and Hieber, 2017).

Document-level models require parallel data with document boundaries for training and evaluation. As compared to sentence-level systems, data for building robust document-level models is significantly low resourced Liu and Zhang (2020). WMT20 provides document-level distinctions for Europarl v9, New-Commentary v14 and Rapid corpus. Our training data is constrained to have only parallel and monolingual data provided by WMT20 shared task, the statistics of data are given in section 4.1. Since, our system is build in Marathi-Hindi direction, we backtranslated Hindi (News Crawl2008-2019) monolingual data into Marathi to generate bitext. This backtranslated data is than concatenated with parallel data made available by organizers, to train machine translation models.

4 Experimental Setup

For our primary submission we use document-level Miculicich et al. (2018) model, an extension of transformer with additional context attentions. For comparison with sentence-based NMT systems, a strong baseline using OpenNMT-py Klein et al. (2017) is first defined. For true comparison, the architecture and configurations of both the models are kept the same.

4.1 Dataset

Table 1 presents details of training, development and test corpus. We used all the parallel data (HI, MR) provided by WMT20 for similar language translation task. The available parallel data was insufficient to train NMT models, therefore we used monolingual "News Crawl" data for generating synthetic parallel corpus through backtranslation. NMT models are trained on backtranslated bitext combined with existing parallel corpus. Training

corpus contains data of multiple domains, a self test set is created by selecting chunk of data from each domain according to size of dataset. Original bitext and backtranslated parallel training data is tokenized with Indic-NLP [3] library, which supports tokenization/de-tokenization of Hindi and Marathi.

Our document-level systems Miculicich et al. (2018) expect document boundaries in text file during training and testing. Available data for this shared task does not contains document boundaries, for this we followed the same approach used by Ul Haq et al. (2020) to generate artificial document boundaries. They have taken average document size from document-level corpora and used the same size to generate document boundaries for parallel data without document distinctions. For train and dev set, instead of splitting on sentences, they considered number of documents. We have used average of two best performing context variables for document size as reported in Table 3 of Miculicich et al. (2018).

Corpus	Sentences	Documents
News	12.3K	4.1K
PmIndia	25.9K	8.6K
IndicWordNet	11.2K	3.7K
NewsCrawl-Monolingual	0.6M	0.2M
Dev	1114	278
Test	1941	485

Table 1: Train, Dev and Test dataset statistics along with document split.

4.2 Model Configurations

As our sentence-level baseline and document-level systems are based on Transformer model, we followed similar configuration parameters for both as reported in original paper Vaswani et al. (2017). 6 hidden layers are incorporated on both encoder and decoder side of Transformer model. All the hidden states have a dropout of 0.1 and 512 dimensions. Transformer model is trained with 8000 warm-up steps with a learning rate of 0.01. We checkpoint the model every 1000 steps for validation. For all the models, batch size is set to 2048 and is trained for 150 epochs.

Two step training process is followed as described by Miculicich et al. (2018). Initially NMT models are optimized without considering contextual information, after that encoder and decoder models are optimized by using context-aware HANs. HAN Transformer models gave best performance for 1-3 previous sentences, we use k=3 previous sentences for both source and target side context.

5 Results

Table 2 shows our results for Hindi−Marathi translations. Our document-level systems for both directions HR⇒MR and MR⇒HI outperformed sentence-level baselines.

BLEU score for WMT, Dev and $Self$ test set is reported in Table 2 for all systems. BLEU score for WMT test data is provided by WMT20 organizers. We have computed BLEU scores using Moses $multi - blue.perl$ script. For submission, we used output of document-level system trained on all data in MR⇒HI direction which gave highest BLEU score (6.79) on WMT test set. Our document-level models are optimized by adding context-aware HANs on encoder side only[4]. With DL−NMT model trained on corpus containing 90% backtranslated data, a gain of 0.63 BLEU points is achieved (6.16 ⇒ 6.79) over sentence-level baseline (row 2).

In last rows (3 and 4) of Table 2, NMT models are build in opposite direction of backtranslated data, depicted as $\text{NMT}_{\text{forward}}$ and and $\text{DL}-\text{NMT}_{\text{forward}}$. For forward translation models, source side is backtranslated data while target side is original monolingual data used for backtranslation. Similarly, DL−NMT models trained in forward direction of data, achieved batter score over NMT systems. Since, the large portion of training data contains synthetic data, on self test set all models performed better due to over fitting.

System	Direction	$BLEUScore$		
		Wmt	Dev	Self
NMT	MR⇒HI	6.16	8.08	12.50
+DL−NMT	MR⇒HI	**6.79**	9.31	14.93
$+\text{NMT}_{\text{fwd}}$	HI⇒MR	3.29	6.33	16.69
$+\text{DL}-\text{NMT}_{\text{fwd}}$	HI⇒MR	3.54	6.28	17.75

Table 2: Table summarizing Document-level NMT (DL-NMT) and NMT Transformer results for different test sets.

[3] https://github.com/anoopkunchukuttan/indic_nlp_library

[4] Due to limited availability of time, HAN for decoder side and HAN joint models were not used for experiments.

6 Summary

This paper presented the "FJWU_NUST" system submitted to the Similar Language Translation task at WMT20. The limited and out-of-domain parallel training data provided by organizers, emerged as a challenging task to train NMT models, whose quality is dependent on large data.

We have utilized monolingual data with back-translation along with available parallel data for training NMT system which incorporated context-aware HANs on encoder side. Our document-level systems outperformed sentence-level NMT systems, even in the absence of document-level corpora. This showed that document-level machine translation can be reasonable alternative of NMT, since it can deliver good quality translation for low-resource languages without requiring document-level parallel data.

7 Acknowledgments

This study is funded by Higher Education Commission of Pakistan's project: National Research Program for Universities (NRPU) (5469/Punjab/NRPU/R&D/HEC/2016).

References

Sadaf Abdul-Rauf, Holger Schwenk, Patrik Lambert, and Mohammad Nawaz. 2016. Empirical use of information retrieval to build synthetic data for smt domain adaptation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(4):745–754.

Loïc Barrault, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. 2019. Findings of the 2019 conference on machine translation (WMT19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 1–61, Florence, Italy. Association for Computational Linguistics.

Rachel Bawden, Rico Sennrich, Alexandra Birch, and Barry Haddow. 2017. Evaluating discourse phenomena in neural machine translation. *arXiv preprint arXiv:1711.00513*.

Junxuan Chen, Xiang Li, Jiarui Zhang, Chulun Zhou, Jianwei Cui, Bin Wang, and Jinsong Su. 2020. Modeling discourse structure for document-level neural machine translation.

Marta R. Costa-jussà. 2017. Why Catalan-Spanish neural machine translation? analysis, comparison and combination with standard rule and phrase-based technologies. In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, pages 55–62, Valencia, Spain. Association for Computational Linguistics.

Anna Currey, Antonio Valerio Miceli Barone, and Kenneth Heafield. 2017. Copied monolingual data improves low-resource neural machine translation. In *Proceedings of the Second Conference on Machine Translation*, pages 148–156.

Tobias Domhan and Felix Hieber. 2017. Using target-side monolingual data for neural machine translation through multi-task learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1500–1505, Copenhagen, Denmark. Association for Computational Linguistics.

Sebastien Jean, Stanislas Lauly, Orhan Firat, and Kyunghyun Cho. 2017. Does neural machine translation benefit from larger context? *arXiv preprint arXiv:1704.05135*.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M Rush. 2017. Opennmt: Open-source toolkit for neural machine translation. *arXiv preprint arXiv:1701.02810*.

Shaohui Kuang, Deyi Xiong, Weihua Luo, and Guodong Zhou. 2017. Modeling coherence for neural machine translation with dynamic and topic caches. *arXiv preprint arXiv:1711.11221*.

Surafel Melaku Lakew, Aliia Erofeeva, and Marcello Federico. 2018. Neural machine translation into language varieties. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 156–164, Brussels, Belgium. Association for Computational Linguistics.

Siyou Liu and Xiaojun Zhang. 2020. Corpora for document-level neural machine translation. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 3775–3781, Marseille, France. European Language Resources Association.

Sameen Maruf and Gholamreza Haffari. 2017. Document context neural machine translation with memory networks. *arXiv preprint arXiv:1711.03688*.

Sameen Maruf and Gholamreza Haffari. 2018. Document context neural machine translation with memory networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1275–1284, Melbourne, Australia. Association for Computational Linguistics.

Lesly Miculicich, Dhananjay Ram, Nikolaos Pappas, and James Henderson. 2018. Document-level neural machine translation with hierarchical attention networks. *arXiv preprint arXiv:1809.01576*.

Nima Pourdamghani and Kevin Knight. 2017. Deciphering related languages. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2513–2518, Copenhagen, Denmark. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Dario Stojanovski and Alexander Fraser. 2020. Addressing zero-resource domains using document-level context in neural machine translation.

Zhaopeng Tu, Yang Liu, Shuming Shi, and Tong Zhang. 2018. Learning to remember translation history with a continuous cache. *Transactions of the Association for Computational Linguistics*, 6:407–420.

Sami Ul Haq, Sadaf Abdul Rauf, Arslan Shoukat, and Noor-e Hira. 2020. Improving document-level neural machine translation with domain adaptation. In *Proceedings of the Fourth Workshop on Neural Generation and Translation*, pages 225–231, Online. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Elena Voita, Pavel Serdyukov, Rico Sennrich, and Ivan Titov. 2018. Context-aware neural machine translation learns anaphora resolution. *arXiv preprint arXiv:1805.10163*.

Longyue Wang, Zhaopeng Tu, Andy Way, and Qun Liu. 2017. Exploiting cross-sentence context for neural machine translation. *arXiv preprint arXiv:1704.04347*.

Hao Xiong, Zhongjun He, Hua Wu, and Haifeng Wang. 2019. Modeling coherence for discourse neural machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7338–7345.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 1480–1489.

Jiacheng Zhang, Huanbo Luan, Maosong Sun, Feifei Zhai, Jingfang Xu, Min Zhang, and Yang Liu. 2018. Improving the transformer translation model with document-level context. *arXiv preprint arXiv:1810.03581*.

Jiajun Zhang and Chengqing Zong. 2016. Exploiting source-side monolingual data in neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545.

Zaixiang Zheng, Xiang Yue, Shujian Huang, Jiajun Chen, and Alexandra Birch. 2020. Toward making the most of context in neural machine translation.

Multilingual Neural Machine Translation: Case-study for Catalan, Spanish and Portuguese Romance Languages

Pere Vergés Boncompte and Marta R. Costa-jussà
TALP Research Center
Universitat Politècnica de Catalunya, Barcelona
pere.verges@est.fib.upc.edu, marta.ruiz@upc.edu

Abstract

In this paper, we describe the TALP-UPC participation in the WMT Similar Language Translation task between Catalan, Spanish, and Portuguese, all of them, Romance languages. We made use of different techniques to improve the translation between these languages. The multilingual shared encoder/decoder has been used for all of them. Additionally, we applied back-translation to take advantage of the monolingual data. Finally, we have applied fine-tuning to improve the in-domain data. Each of these techniques brings improvements over the previous one.

In the official evaluation, our system was ranked 1st in the Portuguese-to-Spanish direction, 2nd in the opposite direction, and 3rd in the Catalan-Spanish pair.

1 Introduction

Research in the field of Machine Translation (MT) has been growing during these last years. From statistical approaches (Koehn et al., 2003) to neural ones (Bahdanau et al., 2015), the progress has been impressive. Even after having achieved exceptional results based only on attention mechanisms (Vaswani et al., 2017), there are still many challenges and improvements remaining, for instance, multilingual translation from languages other than English, which have lower resources, and domain adaptation.

In order to tackle these challenges, the Similar Language Task organized in the context of the Conference on Machine Translation (WMT 2020) has provided an appropriate setting for them. Within this task, the focus is the translation between languages that are different from English, and more specifically, the focus consists of translating languages that are from the same family. The families included are the following: South-Slavic, Indo-Aryan, and Romance.

In our case, we have devoted the research to Romance languages, which include Spanish, Portuguese, and Catalan. The evaluation comprised all translation directions, but only provided parallel training data for Spanish-Portuguese and Spanish-Catalan. We approached the Portuguese-Catalan pair both from a pivot-based and zero-shot perspective.

In this paper, we make use of the well-known multilingual shared encoder/decoder and we show its effectiveness when applied to languages of the same linguistic family. Additionally, we benefited from back-translation and fine-tuning.

2 Background

In this section, we show an overview of neural-based multilingual machine translation and domain adaptation using fine-tuning.

2.1 Multilingual translation

When having multiple languages, there is the opportunity to use several NMT architectures, based in the Transformer (Vaswani et al., 2017). Among the alternatives, we can share encoders and decoders (Johnson et al., 2017) or have specific encoders and decoders for each language (Escolano et al., 2020). In this paper, we are using the shared approach and we are leaving as further work to compare with other ones.

Shared encoder-decoder One direct approach is using a single encoder/decoder shared for all languages (Johnson et al., 2017). In this case, parameters and vocabulary are shared among all language pairs and it helps the generalization across languages improving the translation for the low resource language pairs (Aharoni et al., 2019). Additionally, the shared encoder/decoder allows using zero-shot easily, only by adding a tag in the source sentence. The source sentence has to contain the

Proceedings of the 5th Conference on Machine Translation (WMT), pages 447–450

language abbreviation of the target language. So, when translating from Catalan to Spanish, we have to include the *<2es>* tag at the beginning of the Catalan source sentence, which means that we are translating into Spanish.

```
<2es> Bon dia -> Buenos días
```

Therefore, it is necessary to add the tag to indicate the target language, followed by the sentence to be translated. This is necessary both in training and inference.

2.2 Monolingual corpus selection for back-translation

There is a large amount of monolingual data available for this task. Monolingual data can improve the system by using back-translation (Sennrich et al., 2016). However, back-translation is a process that consumes a lot of resources, so we decided to select the monolingual data within the target domain. The selection criterion has been the TF-IDF (Term Frequency – Inverse Document Frequency), which defines the relevance of the words in a document. Using this criterion, we compared all the available monolingual data against the development set and only kept the files that had a higher score among all.

2.3 Domain adaptation

One approach to improve the translation of a specific language domain is to make use of fine-tuning techniques. Fine-tuning consists of retraining a model that has already been trained with out-of-domain data, with in-domain data. The disadvantage of fine-tuning is that it tends to overfit, due to the small amount of in-domain data used, compared to the out-of-domain data. Sometimes the final model might fall into the problem of catastrophic forgetting (French, 1999).

One approach to avoid over-fitting and catastrophic forgetting is to do mixed fine-tuning, which consists of shuffling the in-domain with the out-of-domain data, and then train normally on this combined data (Chu and Dabre, 2019).

3 Experimental Framework

In this section, we describe the datasets used for the task, the data preprocessing, the training, and the evaluation of the bilingual and multilingual systems.

3.1 Data and Preprocessing

Data Selection All the data used in our experiments has been provided by the organizers, so we did not make use of any additional parallel nor monolingual data. For the Catalan-Spanish and Spanish-Portuguese translation, we used all the parallel data available, which is about 11.3 million sentences for the Catalan-Spanish translation and 4.1 million sentences for the Spanish-Portuguese. For the Catalan-Portuguese we did not have any parallel data. We have also used monolingual data for back-translation purposes. Two million sentences have been used from the *CaWaC* file for Catalan, about 1.1 million sentences from *News-commentary-v15* and *News-crawl-2019* files for Portuguese, and 1.5 million sentences from *News-commentary-v15* and *News-crawl-2015* for Spanish. The multilingual model has been trained using all the parallel data, and with pseudo-parallel data that has been obtained by applying back-translation. To achieve the back-translation we used our best system at the moment to perform the translation of the monolingual data, obtaining the pseudo-parallel corpus. As said in Section 2.2, the monolingual data has been selected using TF-IDF as the measure for text similarity [1]. We used 2/3 of the development set for fine-tuning purposes and 1/3 of the development set as a test set.

Preprocessing We followed the standard procedure for preparing the data, which consists of normalizing, tokenizing, truecasing, and cleaning (limiting sentences from 1 to 50 words). To perform these actions we made use of the *Moses*[2] scripts. We extracted the joint subwords with byte-pair encoding (BPE)[3].

3.2 Parameter Details

The bilingual and multilingual models are both based on the Transformer architecture, implemented with *fairseq* toolkit [4]. We assigned six attention layers for the encoder and the decoder, each having four attention heads per layer, with an embedding dimension of 512. Additionally, all the models shared the source and target embeddings. The multilingual model shared the embeddings among all language pairs. Each batch was

[1] https://github.com/BhargavaRamM/Document-Similarity

[2] https://github.com/moses-smt/mosesdecode

[3] https://github.com/rsennrich/subword-nmt

[4] https://github.com/pytorch/fairseq

assigned to have a maximum number of tokens of 2048. The optimizer used was Adam, setting the betas to β_1 = 0.9 and β_2 = 0.98, with a learning rate of 5e-4 varied with the inverse square root of the step number. The warm-up steps were set equal to 4000, a dropout of 0.1, and a weight decay and gradient clipping norm set to 0.

4 Results

The results show the improvements obtained by applying multilinguality, back-translation, and fine-tuning techniques. For the pair Catalan-Portuguese (CA-PT), in which there was no training data available. We have used the cascade technique, which consists of concatenating the translation of Catalan-to-Spanish and Spanish-to-Portuguese systems, and the other way around for the opposite direction. Also, we have used the multilingual system to obtain zero-shot translation for this pair.

Directions	BI	MULT	+BACK	+FT
ES→CA	64.23	73.12	70.59	71.21
CA→ES	60.64	69.56	73.01	74.05
ES→PT	27.20	27.62	28.80	29.55
PT→ES	29.70	30.57	30.89	32.12
CA→PT	20.99	24.94	25.52	26.94
PT→CA	25.21	28.00	27.97	29.18
CA→PT ZS	-	12.47	13.56	16.05
PT→CA ZS	-	17.67	19.64	19.56

Table 1: BLEU results for all the systems evaluated in the development of this study. BI = Bilingual, MULT = Multilingual, BACK = Multilingual with Backtranslation, FT = Multilingual with back-translation and Fine-tuning, ZS = zero-shot.

Table 1 shows that the multilingual model outperforms the bilingual model in all cases. Zero-shot performs worse than the cascade method. Applying back-translation to the multilingual model improves for most language pairs and directions. Finally, when applying fine-tuning to the back-translation model, we see an improvement in all pairs and directions, except for the PT→CA direction with zero-shot.

4.1 Official evaluation results

Here we report the official evaluation. We participated with our best system which was the multilingual model with back-translation and fine-tuning. For the CA-PT directions, we translated using the cascade technique, Table 2 reports the results on the evaluation test set. Our system was ranked 1st in the Portuguese-to-Spanish direction, 2nd in the opposite direction, and 3rd in the Catalan-Spanish pair. For the Catalan-Portuguese directions, the results were not released.

Directions	BLEU
ES-CA	60.50
CA-ES	68.84
ES-PT	32.33
PT-ES	33.82
CA-PT	32.80
PT-CA	34.40

Table 2: Official BLEU scores for the evaluation of the final test set.

5 Discussion

We will now discuss the results obtained for each system we have trained, comparing one against the others.

Bilingual model compared to the Multilingual model We have shown that the multilingual model outperforms the bilingual model in all translations directions, with an improvement that varies from +0.4 to +6.9 BLEU. The multilingual model allows for a better generalization by sharing the vocabulary among all the languages. Additionally, the multilingual model allows for zero-shot translation.

Back-translation This technique allows us to make use of monolingual data. The improvement with this technique varies from +0.5 to +3.4 BLEU, except when using the monolingual Catalan data (ES→CA and PT→CA directions). This deterioration is probably due to the lower resemblance (estimated using the TF-IDF score) of the *CaWaC* dataset compared to the target domain.

Fine-tuning We have applied fine-tuning to perform the domain adaptation. To do so, we added 2/3 of the development data set to the already trained model, which is the multilingual model with back-translation, since it was the best model we had so far. After doing so, we had to retrain the model from the last checkpoint, preventing it from overfitting. By applying fine-tuning, we were able to achieve improvements between +0.6 and +2.5 BLEU points (except in zero-shot). This fine-tuning improvement is achieved by using very few resources (1500 sentences) and less time compared to back-translation, which requires more resources and time.

6 Conclusion

We have observed how using a multilingual shared encoder/decoder in languages from the same family improves bilingual translation. This is due to a positive transfer among these languages while sharing vocabulary and embeddings. Additionally, this multilingual shared system has been improved with both back-translation and fine-tuning methods.

Acknowledgments

We are grateful to Carlos Escolano for his comments, corrections, and help throughout the investigations. This work is supported in part by the Spanish Ministerio de Ciencia e Innovación, through the postdoctoral senior grant Ramón y Cajal and by the Agencia Estatal de Investigación through the projects EUR2019-103819, PCIN-2017-079 and PID2019-107579RB-I00 / AEI / 10.13039/501100011033

References

Roee Aharoni, Melvin Johnson, and Orhan Firat. 2019. Massively multilingual neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3874–3884, Minneapolis, Minnesota. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Chenhui Chu and Raj Dabre. 2019. Multilingual multi-domain adaptation approaches for neural machine translation. *CoRR*, abs/1906.07978.

Carlos Escolano, Marta R. Costa-jussà, José A. R. Fonollosa, and Mikel Artetxe. 2020. Multilingual machine translation: Closing the gap between shared and language-specific encoder-decoders. abs/2004.06575.

Robert M. French. 1999. Catastrophic forgetting in connectionist networks. *Trends in Cognitive Sciences*, 3(4):128 – 135.

Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proc. of the Conference of the NAACL*, pages 48–54.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

A3-108 Machine Translation System for Similar Language Translation Shared Task 2020

Saumitra Yadav, Manish Shrivastava
Machine Translation - Natural Lanugae Processing Lab
Language Technologies Research Centre
Kohli Center on Intelligent Systems
International Institute of Information Technology - Hyderabad
saumitra.yadav@research.iiit.ac.in
m.shrivastava@iiit.ac.in

Abstract

In this paper, we describe our submissions for Similar Language Translation Shared Task 2020. We built 12 systems in each direction for Hindi $\Longleftrightarrow$ Marathi language pair. This paper outlines initial baseline experiments with various tokenization schemes to train statistical models. Using optimal tokenization scheme among these we created synthetic source side text with back translation. And prune synthetic text with language model scores. This synthetic data was then used along with training data in various settings to build translation models. We also report configuration of the submitted systems and results produced by them.

1 Introduction

Machine Translation systems are models which aim to translate text from one language into another. There are multiple ways of building such a model (Rule Based, Data driven, Hybrid etc.). In this system description paper, we use data driven techniques to build MT systems. As the name suggests, data driven MT systems make use of parallel sentences (i.e. x^{th} sentence in two languages have same meaning). We make use of statistical (Koehn et al., 2003) and neural (Bahdanau et al., 2014) methods to build systems for Hindi Marathi pair.

Hindi Marathi language pair comes under purview of similar languages. Similar Languages are languages which exhibit lexical and structural similarities (Kunchukuttan et al., 2014a). This can be due to common ancestry or being in close proximity for long time. In current digital age communication, translation between similar language is a justifiable requirement. But there is a scarcity of good quality bitext for many language pairs, as is the case of Hindi Marathi. Hence, we used characteristics displayed by similar languages (in this case Hindi and Marathi) like similar form of spelling, pronunciation etc. Following Kunchukuttan and Bhattacharyya (2017) and Kunchukuttan et al. (2014b) we made use of byte pair encoding (Sennrich et al., 2016b) and morfessor toolkit (Virpioja et al., 2013) respectively as part of preprocessing step before training. Using cues from Koehn and Knowles (2017) and looking at the size of training data provided, we use statistical method to build initial models. To further salvage similarity between this language pair we made use of backtranslation (Sennrich et al., 2016a) to generate more synthetic data for further training using both neural and statistical methods.

For this shared task we developed 12 translation systems in each direction (Hindi $\Longleftrightarrow$ Marathi). To rank systems, we went through some test instances subjectively and also compared our BLEU scores with another Translation system. And chose top 2 systems in both direction using both subjective examination and detoknized BLEU scores. Subsequent sections give more detailed overview of systems developed.

2 Seed MT systems using different tokenization schemes

Experiments in Koehn and Knowles (2017) show that Statistical Machine Translation model fairs better when compared to Neural model in case of low resource setting. So, we make use of SMT model to make initial baseline systems using various tokenization schemes. We use these systems as seed system, used to create synthetic dataset for further training by back translation.

2.1 Data

For our initial experiments we just used parallel and monolingual corpora shared by the organizers. We include training data to monolingual corpus for each language (LM corpus) to make language model. Parallel text consisted of bitext from 3

Proceedings of the 5th Conference on Machine Translation (WMT), pages 451–455

Corpus/Language	Hindi				Marathi			
#of Tokens	basicTok	BPE	Morf	#of Sentences	basicTok	BPE	Morf	#of Sentences
Train	840863	977742	38246	38246	638467	867968	851394	38246
Dev	32106	36482	34600	1411	25552	33997	33828	1411
Monolingual	1455510657	1760885875	1629220967	77722389	4834280	6715047	6526439	369403

Table 1: Total number of Tokens in each file after various tokenization schemes, last sub-column in both languages column denotes total number of lines in respective corpus

sources namely *News*, *PM India*, *Indic WordNet*. *Indic Wordnet* is not used in training because we found multiple instances of sentence pairs in which one of the sentence was incomplete.

2.2 Preprocessing

We used the IndicNLP toolkit[1] to tokenize all corpora as first preprocessing step. Then we made use of a BPE (Sennrich et al., 2016b) model trained with 10000 merge operations on the LM corpus for both Hindi and Marathi. The resultant model was used to tokenize words to subwords in sentences for all texts. Morfessor (Virpioja et al., 2013) was also used as another alternative preprocessing step. We trained a morfessor model on the full LM corpus of Marathi and an equally sized Hindi Corpus. And taking cue from IndicNLP toolkit, we used '+' as delimiter when segmenting words into segments i.e. a word *xyz* which was to segment as *x yz* will segment as *x+ yz*. Table 1 shows statistics of preprocessed data. We used all possible combinations of tokenization schemes while training initial models, these tokenization schemes were,

- Basic tokenization denoted as BasicTok in Table 1 which make use of IndicNLP toolkit.
- BPE which tokenize words into subword and is denoted as BPE.
- Tokenization using Morfessor, which is denoted as Morfes.

2.3 Machine Translation Model

We made use of Moses toolkit (Koehn et al., 2007) to build statistical models trained with tokenized bitext. We also use GIZA++ (Och and Ney, 2003) to find alignments between parallel text and grow-diag-final-and method (Koehn et al., 2003) to extract aligned phrases. And utilize KenLM (Heafield, 2011) to train a trigram model with kneser ney smoothing on monolingual corpus of both languages. MERT (Och, 2003) is used for tuning the trained models. We evaluated these models on dev set. Results are given in Table 2.

2.4 Using back-translation to augment training data

Based on the results in Table 2 we make use of following tokenization schemes depending on direction of translation,

- BPE as tokenization preprocessing scheme on both languages when translation direction is from Hindi to Marathi.
- Morf as tokenization scheme for Marathi and Basic tokenization for Hindi when translating from Marathi to Hindi.

After translating monolingual corpus, we did the following post processing based on direction of translation,

- In case of Marathi to Hindi translation, in post processing we remove '+' delimiter. This is due to Marathi being morphological richer than Hindi.
- For Hindi to Marathi,we simply joined the subwords in text translated.

Due to time constraint we translated some part of Hindi monolingual corpus (Authentic$_{Hindi}$) to Marathi (Synthetic$_{Marathi}$). We used beam search with default setting in Moses for this translation. We used already trained LM from Section 2.3 to learn average LM score of BPE tokenized Marathi monolingual corpus. Synthetic$_{Marathi}$ is than pruned (SyntheticPruned$_{Marathi}$) by keeping back-translated sentences which have LM score higher then average LM score on aforementioned Corpus. Same process is followed while translating Authentic$_{Marathi}$ to Synthetic$_{Hindi}$ and further pruning to get SyntheticPruned$_{Hindi}$. Statistics related to back-translated data and resultant pruned corpus is given in Table 3.

[1] http://anoopkunchukuttan.github.io/indic_nlp_library/

Experiment.	Tokenization Based Exp	Hin To Mar	Mar To Hin
1	Hindi BasicTok – Mar BasicTok	19.937	24.542
2	Hindi BPE – Mar BasicTok	19.2251	23.13
3	Hindi Morfes. - Mar BasicTok	19.1327	23.44
4	Hindi BasicTok – Mar BPE	19.02	25.836
5	Hindi BPE – Mar BPE	**20.06**	26.07
6	Hindi Morfes. - Mar BPE	19.43	25.54
7	Hindi BasicTok – Mar Morfes.	19.37	**26.282**
8	Hindi BPE – Mar Morfes.	19.49	25.30
9	Hindi Morfes. - Mar Morfes.	19.33	26.03

Table 2: BLEU Scores on dev dataset when we use SMT models which are trained in all combinations of 3 tokenization schemes.

Back Translation direction L1 to L2	Hindi to Marathi	Marathi to Hindi
Sentences translated	456106	369403
Average KenLM Score of Monolingual data in L2	62.66	42.25
Sentences which are above this LM score	283043 (62.05%)	215417 (58.31%)
Average Sentence length of pruned corpus with standard deviation	10.25, 4.80	11.57, 4.52

Table 3: Statistics of back translated data

3 MT models using augmented bitext

For augmented data experiments we had following datasets available for training,

- Original training text
- Synthetic Marathi and Authentic Hindi
- Synthetic Pruned Marathi and Authentic Pruned Hindi
- Synthetic Hindi and Authentic Marathi
- Synthetic Pruned Hindi and Authentic Pruned Marathi

We ran experiments on following dataset combinations, for Hindi to Marathi Systems with BPE tokenization on both Hindi and Marathi,

1. Original training text + Synthetic Marathi and Authentic Hindi + Synthetic Hindi and Authentic Marathi
2. Original training text + Synthetic Pruned Marathi and Authentic Pruned Hindi + Synthetic Pruned Hindi and Authentic Pruned Marathi
3. Original training text + Synthetic Hindi and Authentic Marathi
4. Original training text + Synthetic Pruned Hindi and Authentic Pruned Marathi

And for Marathi to Hindi System, we ran following dataset combinations with morfessor model tokenization on Marathi and Basic Tokenization on Hindi,

1. Original training text + Synthetic Hindi and Authentic Marathi + Synthetic Marathi and Authentic Hindi
2. Original training text + Synthetic Pruned Hindi and Authentic Pruned Marathi + Synthetic Pruned Marathi and Authentic Pruned Hindi
3. Original training text + Synthetic Marathi and Authentic Hindi
4. Original training text + Synthetic Pruned Marathi and Authentic Pruned Hindi

All these dataset combinations were used to train following methods to build MT models with respective default configurations available in respective toolkits,

- SMT model using Moses toolkit (Koehn et al., 2007)
- NMT model with attention using Opennmt toolkit (Klein et al., 2017)
- NMT model with attention and copy attention (See et al., 2017) using Opennmt toolkit, to make use of similarity between Hindi Marathi language pair

4 Result

To submit two best systems out of 12 in each direction as directed by shared task, we did two evaluations. Firstly, we compared our system outputs to output of another publicly available translation model. Second, we went through some random outputs of all system outputs. We found that in most systems synthetic-authentic dataset which was not pruned with LM scores along with original training set performed better than pruned augmented bitext and original corpus. Following this, we selected following system outputs as our submission,

- Hindi to Marathi System:
  - Primary Submission: NMT with Attention + Original parallel text + SyntheticHindi_AuthenticMarathi
  - Contrastive Submission: NMT with Attention and CopyAttention + Original parallel text + SyntheticMarathi_AuthenticHindi + SyntheticHindi_AuthenticMarathi
- Marathi to Hindi System:
  - Primary Submission: NMT with Attention + Original parallel text + SyntheticMarathi_AuthenticHindi + SyntheticHindi_AuthenticMarathi
  - Contrastive Submission: SMT + Original parallel text + SyntheticMarathi_AuthenticHindi + SyntheticHindi_AuthenticMarathi

Table 4 gives the scores we received for these systems.

Language Direction	Submission Type	BLEU	RIBES	TER
Hindi to Marathi	Primary	11.41	57.2	79.96
Hindi to Marathi	Contrastive	10.21	55.17	82.01
Marathi to Hindi	Primary	18.32	59.31	77.35
Marathi to Hindi	Contrastive	21.11	60.76	77.28

Table 4: Scores for our systems

Both of our Hindi to Marathi Systems were somewhere in the middle compared to the other submissions. On the other hand Marathi to Hindi Contrastive submission (which was trained using SMT) was in top 5 standings.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Kenneth Heafield. 2011. Kenlm: Faster and smaller language model queries. In *Proceedings of the sixth workshop on statistical machine translation*, pages 187–197.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Vancouver, Canada. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*, pages 177–180. Association for Computational Linguistics.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. *arXiv preprint arXiv:1706.03872*.

Philipp Koehn, Franz J Och, and Daniel Marcu. 2003. Statistical phrase-based translation. Technical report, UNIVERSITY OF SOUTHERN CALIFORNIA MARINA DEL REY INFORMATION SCIENCES INST.

Anoop Kunchukuttan and Pushpak Bhattacharyya. 2017. Learning variable length units for SMT between related languages via byte pair encoding. In *Proceedings of the First Workshop on Subword and Character Level Models in NLP*, pages 14–24, Copenhagen, Denmark. Association for Computational Linguistics.

Anoop Kunchukuttan, Abhijit Mishra, Rajen Chatterjee, Ritesh Shah, and Pushpak Bhattacharyya. 2014a. Sata-anuvadak: Tackling multiway translation of indian languages. *pan*, 841(54,570):4–135.

Anoop Kunchukuttan, Ratish Pudupully, Rajen Chatterjee, Abhijit Mishra, and Pushpak Bhattacharyya. 2014b. The iit bombay smt system for icon 2014 tools contest. *NLP Tools Contest at ICON 2014*.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st annual meeting of the Association for Computational Linguistics*, pages 160–167.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational linguistics*, 29(1):19–51.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Sami Virpioja, Peter Smit, Stig-Arne Grönroos, Mikko Kurimo, et al. 2013. Morfessor 2.0: Python implementation and extensions for morfessor baseline.

The University of Maryland's Submissions to the WMT20 Chat Translation Task: Searching for More Data to Adapt Discourse-Aware Neural Machine Translation

Calvin Bao* **Yow-Ting Shiue*** **Chujun Song** **Jie S. Li** **Marine Carpuat**

Department of Computer Science, University of Maryland, College Park, MD

{csbao,ytshiue,cjsong,jli2718,marine}@cs.umd.edu

Abstract

This paper describes the University of Maryland's submissions to the WMT20 Shared Task on Chat Translation. We focus on translating agent-side utterances from English to German. We started from an off-the-shelf BPE-based standard transformer model trained with WMT17 news and fine-tuned it with the provided in-domain training data. In addition, we augment the training set with its best matches in the WMT19 news dataset. Our primary submission uses a standard Transformer, while our contrastive submissions use multi-encoder Transformers to attend to previous utterances. Our primary submission achieves 56.7 BLEU on the agent side (en→de), outperforming a baseline system provided by the task organizers by more than 13 BLEU points. Moreover, according to an evaluation on a set of carefully-designed examples, the multi-encoder architecture is able to generate more coherent translations.

1 Introduction

Recent advances have made MT a widespread tool for asynchronous consumption of text. The dream of dissolving language barriers, however, will not be fulfilled until MT enables two or more people carry on a synchronous conversation, each speaking their native languages. Building translation systems that enable seamless conversations between an English-speaking customer support agent and a German-speaking customer is the goal of WMT20's shared task of chat translation (Farajian et al., 2020). In participation of this shared task, we focused on the agent side, translating English utterances into German. Our methods are inspired by Voita et al. (2018) and Bawden et al. (2018), explicitly leveraging broader context to address coreference and cohesion to improve translation quality. We compare architectures of a standard transformer with a single encoder and a multi-encoder one with an additional transformer encoder to incorporate information from the previous utterance. In the case of blind testing or production use, since customer target utterances (English) will not be given, a separate de→en model was trained and used to back-translate customer utterances.

Additionally, given the limited training pairs, we experiment with augmenting our dataset. We selected a subset of WMT19 en-de news data that were similar to the chat training data, which we then added to the training data. The subset was constructed using a full-text search engine loaded with the entire en-de WMT19 news data, which iterated through each chat training example, querying for the two closest matches with both the source and target as search strings.

Our primary system, denoted `PRIMARY`, is a single-encoder pretrained transformer fine-tuned on WMT20 Chat data. The first contrastive system, denoted `CONTRASTIVE1`, is a multi-encoder transformer that pre-warms, using WMT19 news data, the weights of an additional encoder after loading the pretrained transformer. The second contrastive system, denoted `CONTRASTIVE2`, is a multi-encoder transformer that fine-tunes the pretrained transformer on a combination of WMT19 news data and WMT20 chat data.

2 Related Work

One of the main challenges for translating discourse arises from ambiguities of sentences when they are taken out of context, as MT models often do (Yamashita et al., 2009). Especially in dialogue, sentences tend to reference entities in previous sentences, which necessitates using cross-sentential information to translate a given sentence. Individual words can be translated in different ways,

*Equal contribution.

Proceedings of the 5th Conference on Machine Translation (WMT), pages 456–461
Online, November 19–20, 2020.

significantly varying the meaning of the resulting sentence in a larger context (Gao et al., 2015). In addition, dialogue in the customer support domain is a distinctive and spontaneous category of text, with colloquialisms, errors and minimal revisions. All of these deviations can accumulate error throughout the course of a conversation.

Dialogue Translation: Specific interests in translating dialogues can be found as early as Lee and Kim (1997)'s work on Korean-English dialogue translation based on syntactic patterns and *n*-grams. Though their model parses sentences into speech acts instead of generating full-sentence translations, they have pointed out the importance of context (previous sentence) in interpreting the current sentence properly. The most relevant recent work is (Maruf et al., 2018), in which contexts for both source-side and target-side are utilized as additional generation conditions for the decoder in their NMT model. Several variants of the model architecture and the attention mechanism are explored. However, their experiments are conducted on Europarl and OpenSubtitles. The former is formal language and the latter scripted conversations of movies and TV. Here, in contrast, chat data is informal unscripted real-world language.

Context-Aware Machine Translation: Chat translation can be regarded as a special case of context-aware translation. Jean et al. (2017) extends the vanilla attention-based neural MT model (Bahdanau et al., 2015) by conditioning the decoder on the previous sentence via attention over its words. Wang et al. (2017) propose a cross-sentence context-aware model. They integrate the historical representation into NMT with two strategies: a warm-start of encoder and decoder states, and an auxiliary context source for updating decoder. Bawden et al. (2018) use multi-encoder NMT models to exploit context from the previous source and target sentence. Voita et al. (2018) propose a context-aware model based on the Transformer. Their model controls the flow of information from the extended context and improves on pronoun translation.

NMT Facilitated with Retrieved Translations: There is a line of NMT research inspired by example-based translation systems that aims to generate better translations by retrieving and referencing additional translation pairs. Gu et al. (2018) utilize an off-the-shelf search engine to retrieve training sentence pairs whose source side is similar to a given source sentence and incorporate them as additional input to the decoder. Zhang et al. (2018) use the retrieved examples at prediction time to upweight outputs whose constituents match retrieved *n*-gram translation candidates. In a similar vein, but at training time rather than prediction time, we use a retrieval system to select similar examples from a larger dataset to augment the smaller in-domain training set.

3 Data Preparation

3.1 Preprocessing

We used the Moses toolkit (Koehn et al., 2007) to preprocess our data. The training corpus was tokenized and cleaned. After that, we applied byte pair encoding (BPE) (Sennrich et al., 2016) on the data with the BPE model learned on the data of the pretrained model (Section 4.1.1). Following the pre-trained model, we use its shared vocabulary for both target and source sides. The size of the vocabulary, which is the union of English and German tokens, is 36,628.

3.2 Retrieval-Based Training Data Augmentation

There were only 13.85k utterances in the provided parallel WMT20 Chat training data. Given the limited data, we start off with a model pretrained on the WMT17 en-de news data, and additionally augment our training data with a filtered set of 4.75k lines of WMT19 en-de news data. We adopted Elasticsearch[1] to build a fast full-text search engine on the entire WMT19 en-de news set, and then iterated through each (source, reference) pair in the Chat training data. With each pair, we used the search engine to find the top two matches with the current source and target as search strings. We truncated this set to 4.75k training samples to limit the possibility of overwhelming our fine-tuning set and denote it *Chat-Similar News*. This technique brings the total training set to 18.6K parallel utterances.

4 Experiments

We conducted varied experiments in the English-German direction. We included the English reference of the customer utterances as training data for the scope of these experiments, even though this would not be available in a production setting. This was a strategy to provide more training pairs to our

[1] https://www.elastic.co/

models, knowing that the English references for customers is natural language, according to task organizers.

4.1 Systems Overview

We base all our systems off the Transformer architecture. Our implementation is based on the JoeyNMT toolkit (**?**). We kept hyperparameters common throughout. We experimented with the following settings:

1. Trained a standard single-encoder Transformer model.

2. Introduced a second encoder into our NMT architecture to process the preceding sentence, using context-target attention along with source-target attention to compute the final encoder hidden state, on a combination of *Chat-Similar News* and Chat.

3. As in item 2, introduced a second encoder and pre-warmed that encoder's weights on *Chat-Similar News*, before fine-tuning on Chat.

4.1.1 Off-the-shelf Pretrained Model

We found that an existing model trained on a different domain can generalize to this smaller dataset. We downloaded model weights for WMT17 en-de, provided by JoeyNMT Transformer[2]. This model was able to adapt to the chat domain, so the pretrained model was used for all experimental settings.

4.1.2 Common Hyperparameters

We kept hyperparameters consistent across models we tested, with some exceptions to account for slight differences in architecture. All models had embedding and hidden layers with 512 units, and feed-forward layers with 2048 units. A dropout rate of 0.1 was used on both the encoder and decoder layers. Training was performed with the Adam optimizer and in minibatches of 2048 tokens, with cross-entropy loss, an initial learning rate of 0.0002, and a patience of 8 validation cycles. All models were trained for a maximum of 65 epochs. The checkpoint with lowest validation perplexity is selected as the final model. For all validation cycles, greedy decoding is adopted. For testing, we used beam search decoding with a beam width of 5.

[2] https://www.cl.uni-heidelberg.de/statnlpgroup/joeynmt/wmt_ende_transformer.tar.gz

4.2 Single-encoder Implementation: `PRIMARY`

We trained a discourse-agnostic Transformer model with self-attention. This model had 6 layers for the both the encoder and decoder, each with 8 attention heads. A single-encoder implementation fine-tuned only on the Chat data was used to produce the primary submission results. We selected this model due to its slightly higher Chat validation BLEU (Table 1). It also achieves the highest test BLEU but with only minor differences with the contrastive systems. However, the gaps between it and the two contrastive multi-encoder implementations are not wide as can be seen.

4.3 Multi-encoder Implementation

Two context-aware models that are partial extensions of that described in Voita et al. (2018) were produced for the contrastive submissions. Voita et al. (2018)'s context-aware model encodes a source sentence and a context sentence independently and applies a gating function to produce a context-aware representation of the source sentence. We explored this combination idea by implementing a trainable gating function, *à la* (Voita et al., 2018), that takes the independently encoded source-side context and independently encoded source-side sentence as inputs to generate a representation for the decoder. Each layer retained 8 attention heads. We used 6 layers in each encoder. The total number of trainable parameters can be seen in Table 2.

4.3.1 Incremental Domain Adaptation: `CONTRASTIVE1`

This system has two steps: we pre-warm the context encoder of a multi-encoder implementation by fine-tuning on *Chat-Similar News* and validating on a subset of the Chat training data. We then fine-tune this intermediate model on the Chat training data, validating against Chat validation data. We consider this an incremental domain adaptation technique because we prewarm the trainable parameters of a new encoder with similar data, before finally tuning on the Chat data. Compared to a multi-encoder baseline implementation trained strictly on Chat, we achieve a 1.79 validation BLEU point improvement. Compared to `CONTRASTIVE2`, a model that fine-tunes on a mixture of Chat and *Chat-Similar News* in one step, we achieve a 0.59 validation BLEU point improvement.

System	Architecture	Domain Adaptation	Dev. BLEU	Test BLEU	Human Score
BASELINE	Standard Transformer	Chat	-	43.4	-
PRIMARY	Standard Transformer	Chat	**58.54**	56.7	79.29
Vanilla Chat	Multi-encoder	Chat	57.52	-	-
CONTRASTIVE1	Multi-encoder	Similar News → Chat	58.31	55.6	-
CONTRASTIVE2	Multi-encoder	Chat + Similar News	57.72	56.4	-
WMT20-CHAT-BEST	-	-	-	**60.1**	**88.21**

Table 1: Agent-side (en→de) performance of submitted systems on the official development and test sets of the WMT20 chat translation task. BASELINE was the best performing model in the WMT19 News task, PRIMARY is our primary submission, and WMT20-CHAT-BEST produced the best Agent-side outputs, according to human evaluation.

Model	# Params	# Samples
PRIMARY	63M	13845
CONTRASTIVE1	83M	(1303, 13845)
CONTRASTIVE2	83M	18624

Table 2: # Trainable parameters and # Training samples per model. Values within tuples indicate the number of training samples available to a corresponding, intermediate model.

4.3.2 Same-time Training of Chat-Similar News: CONTRASTIVE2

This system fine-tunes on the multi-encoder architecture with the combined *Chat-Similar News* and Chat training data in one shot. We see only a 0.2 validation BLEU point improvement here over the multi-encoder fine-tuned only with Chat (Vanilla Chat in Table 1).

5 Evaluation

5.1 Official Evaluation

BLEU scores on the development and test sets, and official human evaluation results are shown in Table 1. The PRIMARY system achieves the best validation and test BLEU. While CONTRASTIVE1 has a slightly higher validation BLEU, it turns out that CONTRASTIVE2 performs better at test time, showing that the same-time training technique may be less prone to overfitting.

5.2 Coherence Evaluation with Hand-crafted Examples

The official evaluation results seem to suggest the context-aware multi-encoder architecture (contrastive systems) is not superior to the standard Transformer which has no access to contextual information. We manually examined the training data, and noted that between two people interacting with each other on the phone or through their computer screens, there are not many indirect pronouns, possibly because there is no associated real-life gesturing necessitating expressions such as "that one" or "those ones". Seemingly, in the provided datasets, the need to be clear over the phone/internet means key words are often repeated for clarity, **especially** on the agent side ("I would like to order a pizza"; **"how can I help you with ordering a pizza"**). Inspired by (Bawden et al., 2018), we carefully evaluate performance of the systems on a hand-crafted dataset consisting of coreference and cohesion test instances. Example instances can be seen in Tables 4 and 5 respectively.

A contributor fluent in both English and German produced two versions of a dataset of 103 source-target pairs[3] based loosely off the provided validation set, following the spirit of (Bawden et al., 2018), in which a current utterance will require the previous utterance in order to make a disambiguating translation in the current. One version has the reference translation set to the correct coreference or cohesion resolution, while the other version can be a potentially correct translation viewing the source sentence in isolation but is incorrect with the additional context. The source side remains unchanged in both versions. We benchmarked each of our submitted models by producing hypotheses using each model given the source sentence, and then computing BLEU scores on the reference from both versions of this dataset.

In Table 3, we show the results of each model against the two versions. We used greedy decoding to generate the hypotheses. We observe that the contrastive multi-encoder systems, though performing worse in BLEU than the single-encoder primary system on the provided validation dataset, actually score higher in the specifically crafted correct coreference/cohesion dataset. By contrast, PRIMARY scores higher for the incorrect coreference/cohesion dataset. Furthermore, the difference in BLEU points between the correct and incorrect

[3] https://github.com/SongChujun/joeynmt/blob/master/chatnmt/coher/manual_coher.json

System	BLEU with Correct Ref.	BLEU with Incorrect Ref.	Diff
PRIMARY	50.44	**49.54**	-0.90
CONTRASTIVE1	50.64	48.84	-1.80
CONTRASTIVE2	**51.39**	49.23	-2.16

Table 3: Agent-side (en→de) performance of submitted systems on our coherence dataset.

Context utterance	Nein. Ich weiss nicht wo **sie** ist. (*No, I do not know where it is.*)
Source utterance	It's 200 meters north of City Center.
Correct reference	**Sie** ist zweihundert Meter nordlich vom Stadtzentrum.
Incorrect reference	**Es** ist zweihundert Meter nordlich vom Stadtzentrum.

Table 4: Example of sentence requiring anaphoric pronoun resolution. A better translation should bias to the correct pronoun based on context as 'sie' and not as 'er' or 'es' (for masculine and neuter nouns respectively).

Context utterance	Ist 30% in Ordnung als Trinkgeld? (*Is 30% alright as tip?*)
Source utterance	Yes, that's more than **generous**.
Correct reference	Ja, das ist mehr als **grosszuegig**.
Incorrect reference	Ja, das ist mehr als **wohlwollend**.

Table 5: Example of sentence requiring lexical disambiguation. Given the context of giving a "tip", a system should bias the translation of "generous" more towards "grosszuegig" (someone is free with money) and away from "wohlwollend" (more in the altruistic, do-gooder sense), which is inappropriate here.

coherence datasets is more significant in the contrastive systems, suggesting that the contrastive models are recovering more of the correct coreference and cohesion, as opposed to retrieving vocabulary words in other areas of the reference.

6 Discussion

Our results largely agree with those of (Voita et al., 2018), chiefly that combining knowledge from a previous "context" sentence can improve the model's ability to improve translation quality when measured against sentences whose translations require anaphora considerations. To accommodate this, we produced one set of sentences which require coreference and cohesion resolution, and one set of sentences that have invalid resolution. We found that each submitted system scored worse on the invalid set compared to the valid set, but the difference was more staggering (Table 3) in the context-aware contrastive systems, lending evidence that these models are able to resolve this type of anaphora.

Our work and submission to the shared task can be viewed with several caveats in mind, which may explain the sub-optimal performance of the contrastive systems compared to the primary system. First, we used hyperparameters consistent with a context-agnostic pretrained model in order to have a fair comparison for evaluation and because these presumably have been well-tuned *for the original model*. It may be the case that different hyperparameters would work better for this particular data and the slightly larger architectures used for the contrastive submissions. It would be worth strategizing with better hyperparameter optimization.

Secondly, we use the provided target sides of the de→en direction to provide context to our en-de data as if it were back-translated. Since both the agent and customer sides of this datasets were actually produced in English (the latter being translated with human-corrected machine translation), these additional utterances are likely higher quality than we would get from back-translating in a real test setting.

7 Conclusions

In this paper, we discussed our methods for training and submitting the outputs of three models for the WMT20 shared task of chat translation. Each system was based off a transformer model pretrained on WMT17 en-de news to provide better fluency. Our best system achieves a test BLEU of 56.7, improving over the provided baseline by more than 13 BLEU points, and less than 4 points behind the best shared task submission. Though we were unable to show that a context-aware model produced better translation quality than the context-agnostic model on the given dataset, our coherence evaluations indicated that it can produce better translations when measured against references needing context for

coreference and cohesion resolution. This was validated both in terms of BLEU and by model scoring of references.

Acknowledgments

This work was partially supported by AWS Cloud Credits provided by an Amazon Machine Learning Research Award. We also thank the Natural Language Processing Laboratory, National Taiwan University[4], where a contributor was previously affiliated, for partially supporting us in computational resources.

References

Dzmitry Bahdanau, K. Cho, and Y. Bengio. 2015. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Rachel Bawden, Rico Sennrich, Alexandra Birch, and B. Haddow. 2018. Evaluating discourse phenomena in neural machine translation. *ArXiv*, abs/1711.00513.

M. Amin Farajian, António V. Lopes, André F. T. Martins, Sameen Maruf, and Gholamreza Haffari. 2020. Findings of the wmt 2020 shared task on chat translation. In *Proceedings of the Fifth Conference on Machine Translation*.

Ge Gao, Bin Xu, David C Hau, Zheng Yao, Dan Cosley, and Susan R Fussell. 2015. Two is better than one: improving multilingual collaboration by giving two machine translation outputs. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*, pages 852–863.

Jiatao Gu, Yong Wang, Kyunghyun Cho, and Victor OK Li. 2018. Search engine guided neural machine translation. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

S. Jean, Stanislas Lauly, Orhan Firat, and K. Cho. 2017. Does neural machine translation benefit from larger context? *ArXiv*, abs/1704.05135.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Jae-won Lee and Gil Chang Kim. 1997. A dialogue analysis model with statistical speech act processing for dialogue machine translation. In *Spoken Language Translation*.

Sameen Maruf, André F. T. Martins, and Gholamreza Haffari. 2018. Contextual neural model for translating bilingual multi-speaker conversations. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 101–112, Brussels, Belgium. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Elena Voita, Pavel Serdyukov, Rico Sennrich, and Ivan Titov. 2018. Context-aware neural machine translation learns anaphora resolution. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1264–1274, Melbourne, Australia. Association for Computational Linguistics.

Longyue Wang, Zhaopeng Tu, Andy Way, and Qun Liu. 2017. Exploiting cross-sentence context for neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2826–2831, Copenhagen, Denmark. Association for Computational Linguistics.

Naomi Yamashita, Rieko Inaba, Hideaki Kuzuoka, and Toru Ishida. 2009. Difficulties in establishing common ground in multiparty groups using machine translation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 679–688.

Jingyi Zhang, Masao Utiyama, Eiichro Sumita, Graham Neubig, and Satoshi Nakamura. 2018. Guiding neural machine translation with retrieved translation pieces. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1325–1335, New Orleans, Louisiana. Association for Computational Linguistics.

[4] http://nlg.csie.ntu.edu.tw/

Naver Labs Europe's Participation in the Robustness, Chat, and Biomedical Tasks at WMT 2020

Alexandre Bérard **Vassilina Nikoulina** **Ioan Calapodescu** **Jerin Philip***
first.last@naverlabs.com
Naver Labs Europe
IIIT Hyderabad

Abstract

This paper describes Naver Labs Europe's participation in the Robustness, Chat, and Biomedical Translation tasks at WMT 2020. We propose a bidirectional German ↔ English model that is multi-domain, robust to noise, and which can translate entire documents (or bilingual dialogues) at once. We use the same ensemble of such models as our primary submission to all three tasks and achieve competitive results. We also experiment with language model pre-training techniques and evaluate their impact on robustness to noise and out-of-domain translation. For German, Spanish, Italian, and French to English translation in the Biomedical Task, we also submit our recently released multilingual *Covid19NMT* model.

1 Introduction

We participate in three German ↔ English tasks: Robustness, Chat, and Biomedical. Because these tasks allow the use of the same German-English data, we are able to submit a single model to all of them. We use adapter layers (Bapna and Firat, 2019) to specialize this common model on the provided in-domain data, and obtain a single multi-domain model.

1.1 Task description

Robustness Task This task is split into two tracks: 1) a *zero-shot* translation track whose goal is to make NMT models that are robust to unseen domains; 2) a *few-shot* translation track, where only a few thousand examples of a new domain will be provided as training data, to try to improve translation quality on this particular domain, while maintaining good quality on the other domains.

Chat Translation Task The goal of this task is to translate bilingual customer dialogues between two participants (one German-speaking "customer" and one English-speaking "agent") to the language of the other participant. It combines three challenges: document-level translation (of dialogues), domain adaptation, and noise robustness. Note that the data was originally all in English (even the customer side) and human-translated to German.

Biomedical Task This task is a typical domain adaptation task, where we have access to large amounts of generic parallel data, and smaller amounts of in-domain data. The provided test sets are at the document level, which may be useful to our document-level approach. While the data is clean, it contains many numbers, named entities, and compound medical terms, which may require some "robustness tricks" to handle properly.

Note that this task is very à propos, considering the current pandemic situation, in which a good-quality biomedical MT model could be very helpful for translating guidelines, news articles about COVID-19, or social media reactions. So, in addition to submitting our German ↔ English multi-domain model, we also participate in several language pairs (German, Spanish, Italian, and French to English) with our recently released multilingual *Covid19NMT* model (Bérard et al., 2020).[1]

1.2 Data

Table 1 describes the training data we used to train our models. The domain-specific training data (BConTrasT, Medline, and Robustness few-shot) was only used to fine-tune model instances for the relevant tasks. We filtered all the training data based on length (min 1 token, max 200, max ratio of 1.8), and automatic language identification with

*Work done during the author's internship at Naver Labs Europe

[1]This model can be downloaded here: https://github.com/naver/covid19-nmt

Proceedings of the 5th Conference on Machine Translation (WMT), pages 462–472
Online, November 19–20, 2020.

`langid.py` (Lui and Baldwin, 2012). We also removed duplicate sentence pairs.

We filtered the Medline training data to remove any sentence pair where either side (English or German) was in the Medline 2018 test sets so that we can use *Medline-test2018* for early stopping.

The *Covid19NMT* model (Bérard et al., 2020) used for our Spanish, Italian and French to English submissions to the Biomedical Task was trained on much larger amounts of training data, obtained from WMT and OPUS (Tiedemann, 2012).[2] It was trained in a multilingual way (many-to-one) with general-domain as well as biomedical data using domain tags (Kobus et al., 2017).

Table 2 describes the validation and test data we used. Some test sets, like *newstest2019* and *Medline-test2019* were only used for the final evaluation in this paper, while others (*BConTrasT-dev* and *Medline-test2018*) were also used for early stopping and model selection.

Corpus	Sents	Docs
Paracrawl	33.9M	–
Rapid2019	965k	48.3k
Europarl	1.75M	6.7k
Commoncrawl	1.97M	–
Wikimatrix	5.68M	–
Wikititles	176k	–
News-commentary	352k	9.1k
News-crawl (de)	440M	20.7M
News-crawl (en)	269M	10.9M
BConTrasT (Chat)	13845	550
Medline (Biomedical)	34710	3452
Robustness few-shot	8503	–

Table 1: Training data size (in number of sentence pairs, and document pairs when available). News-crawl corpora are monolingual.

Corpus	Sents	Docs
newsvalid (de-en)	4499	222
newsvalid (en-de)	4502	185
newstest2019 (de-en)	2000	145
newstest2019 (en-de)	1997	123
IT-valid	1000	–
QED-valid	1117	–
BConTrasT-dev	1902	78
Medline-test2018 (de-en + en-de)	656	96
Medline-test2019 (de-en)	573	50
Medline-test2019 (en-de)	619	50

Table 2: Validation corpora. *IT-valid* is the validation data of the WMT16 IT translation task (*Batch3a*). *Medline-test2018* is the concatenation of WMT18 Biomedical task's *Medline* test sets for *de-en* and *en-de* (as they are too small individually). *newsvalid* is the concatenation of the 2016, 2017 and 2018 News Task test sets, split into two halves: German-original (*de-en*) and English-original (*en-de*).

2 Our model

We explore several techniques to train a model that should be able to cover all tasks with minimal adaptation. We want our model to be bidirectional, robust to noise and new domains, and to be able to translate full bilingual documents at once.

2.1 Pre-processing

We normalize all whitespaces and apply Moses' `deescape-special-chars.perl` on the training data (Koehn et al., 2007).

We train a joint BPE model on the general-domain WMT20 parallel data (English plus German) with 24k merge operations and inline casing, which improves robustness to capitalized inputs (Bérard et al., 2019). We use an in-house BPE implementation similar to `SentencePiece` (Kudo and Richardson, 2018). Like the latter, it does Unicode NFKC normalization and pre-tokenizes its inputs based on their script. It also segments numbers and punctuation character-by-character. We only keep single characters in the dictionary whose count in the training data is greater than 1000. Rarer characters are replaced by a `<copy>` placeholder if they appear on both sides, and an `<unk>` token if they appear only on the source side. We drop them if they are on the target side only. At test time, we can decide whether an OOV character should be copied or ignored, by replacing it with `<copy>` or `<unk>`.[3] We choose to copy *unicode symbols* (including emojis and math symbols) and to ignore the other characters.

We start each source sentence with a source language tag and each target sentence with a target language tag. For documents, each sentence is prefixed with a language code, effectively acting as a sentence delimiter. In the Chat translation task, the language code is also an easy way for the model to detect the current speaker.[4]

[2] Contrary to the other tasks, the Biomedical Task puts no constraint on the training data used.

[3] Copy is followed by a post-processing step, where we replace target-side `<copy>` tokens by the source-side OOV symbols in the same order.

[4] Even though using these tags is not necessary for sentence-

We modified fairseq to load and pre-process its training data on the fly (normalization, BPE, tagging, synthetic noise, binarization, and batching). The advantage of this approach over the statically pre-processed training sets is that we can easily apply a different pre-processing at each epoch. This is useful for BPE dropout (where ideally, we'd like a different segmentation at each epoch) and for noise generation. We can also more easily sample from multiple corpora, and subsample from parallel documents or randomly create fake documents.

We train a sentence-level bidirectional model on the concatenated German → English and English → German parallel data (about 90M examples total), which we use as a baseline for the next steps.

2.2 Pre-trained encoder

Previous works (Edunov et al., 2019; Conneau and Lample, 2019; Clinchant et al., 2019; Lewis et al., 2020; Rothe et al., 2020) show that pre-trained LMs can improve performance of NMT models, especially in low-resource settings. Clinchant et al. (2019) show that, even though the benefit of pre-trained LM is less clear in high-resource settings, it can lead to better domain robustness. In this work, we explore this aspect further and experiment with several pre-trained models for encoder initialization. First, we train a Masked Language Model (MLM) that follows the same architecture as the NMT model's encoder. Since our encoder is bilingual (it encodes both English and German) we train the MLM on a concatenation of large monolingual English and German datasets (100M lines in total per language from news-crawl, news-discuss, and Common Crawl).

We also experiment with a large publicly available MLM model: RoBERTa Base,[5] and initialize our NMT encoder with this model's parameters. Then, we train all parameters further on the NMT task. Using an existing model saves us the cost of having to train a new model. But there are a few downsides: RoBERTa is English-only, so we cannot use it in a bidirectional setting. We are also constrained to use RoBERTa's tokenizer and vocabulary, which prevents us from sharing source and target embeddings. It also complicates custom source-side pre-processing techniques (e.g., inline casing). Our models initialized with RoBERTa have a separate target-side (German) vocabulary of size 24k. They do not use any of the tricks (no copy symbol, inline casing, back-translation, etc.)

Previous work (Voita et al., 2019; Tenney et al., 2019) suggests that the last layers of a pre-trained LM might not be useful for the final task. For this reason, we also try initializing the encoder with the first 8 (out of 12) layers from RoBERTa.

2.3 Tagged back-translation

We back-translate the German and English *news-crawl* monolingual corpora (see Table 1) using our bidirectional Transformer Big baseline with sampling (Edunov et al., 2018). Back-translation is done at the sentence level, but we reassemble the output sentences and their corresponding sources into pairs of documents for document-level training. Like Caswell et al. (2019); Bérard et al. (2019), we prefix the back-translated examples with `<BT>`.

We downsample from our training corpora so that an epoch always corresponds to roughly 90M samples[6] regardless of the presence of back-translation or document-level training; and so that real and back-translated data are approximately balanced. We also upsample the real document pairs by a factor of 100, as we expect them to be more valuable to document-level training than fake documents and back-translated ones.[7]

2.4 BPE dropout

Kudo (2018) propose "subword regularization", a non-deterministic tokenization algorithm, whose stochasticity level can be controlled thanks to a probability parameter. They show that using it to encode the training data acts as regularization and that it can improve translation quality for low-resource or out-of-domain translation. Provilkov et al. (2020) implement the same idea with the BPE algorithm, which they call "BPE dropout".

We apply BPE dropout over the source side of the training data with probability 0.1, as our early experiments with target-side BPE dropout gave worse results than regular BPE.

level models, we wanted our sentence-level and document-level models to share the same pre-processing so that we could easily combine them in ensembles if need be.

[5] https://github.com/pytorch/fairseq/tree/master/examples/roberta

[6] A "sample" being a sentence pair in sentence-level training, or a pair of documents (real, sub-sampled or fake) in document-level training.

[7] For instance, when training document-level bidirectional models with back-translation, an "epoch" consists in 41.7M pairs of fake documents, 42.5M pairs of back-translated documents, and 6M pairs of real documents.

2.5 Noise generation

To increase robustness to noise, we inject random synthetic noise on the source side of our training data (Belinkov and Bisk, 2018; Karpukhin et al., 2019; Vaibhav et al., 2019; Bérard et al., 2019). We modify each sentence with probability 0.1, and each character within this sentence with probability 0.1. Character modifications are either a deletion, a swap with the next character, a duplication, a substitution with a random candidate character, or a character insertion at the preceding position. Candidate characters are extracted from the model's German-English dictionary and sampled according to their rank in this dictionary using a Zipf distribution. Like for back-translation, we start each noised source sequence with a special `<noisy>` tag. Thanks to our on-the-fly pre-processing, we generate new noise at each epoch.

2.6 Document-level training

Like Junczys-Dowmunt (2019); Saleh et al. (2019) we train our models on parallel documents of size up to 1024 BPE tokens. Table 1 sums up the available parallel corpora with document boundaries. We use similar techniques as Junczys-Dowmunt (2019):

- All parallel documents are randomly subsampled into smaller documents (of consecutive sentences).
- The sentence-level parallel data (e.g., ParaCrawl) is also used and transformed into fake documents by randomly merging consecutive sentences.[8] The source side of these documents is prefixed with `<fake>`.

We also keep the same techniques as before: back-translation, BPE dropout, and noise. They just work on full documents instead. To deal with potentially noisy and bilingual documents, we also do the following:

- Each parallel document (including the fake ones) has a 0.2 probability of having all its source/target sentences randomly swapped. The goal is to have an MT model that can translate bilingual documents.
- We randomly drop each sentence delimiter on both the source and target side with probability 0.1. The goal is to force the model to rely exclusively on the source-side delimiters for generating output delimiters, and not on end-of-sentence punctuation. We hope that this will help generate documents of the same length as the input documents.

2.7 Domain adaptation

For domain adaptation, we test two settings: fine-tuning the entire model on the in-domain data (Freitag and Al-Onaizan, 2016), or adding domain-specific adapter layers which we train while freezing the other parameters (Bapna and Firat, 2019; Philip et al., 2020). While fine-tuning is often the optimal strategy, adapters can achieve close performance while significantly reducing the number of parameters per task: we can have a single model for all tasks, with a small set of additional parameters for each task.

We use adapters of size 64 and 1024 respectively for sentence-level and document-level models.[9] We found that the sentence-level models quickly overfit the in-domain data when trained with higher capacity adapters. When fine-tuning the whole model, we continue with the same learning rate schedule as the pre-trained model. When training adapters, we use a fixed learning rate of 10^{-4} and train on a single GPU without delayed updates.

For domain adaptation, we disable noise generation, BPE dropout, and fake documents. When possible, domain adaptation of the document-level models is done with document-level in-domain data. Early stopping is done according to document-level perplexity on the validation sets.

For the Chat Translation Task, we include the training data in both the forward and backward directions (i.e., target side as source and source side as target). We prefix backward sources with the `<BT>` tag. For the other tasks, we adapt the bidirectional models with the in-domain data in both directions if available (i.e., our adapters are bilingual).

3 Experiments

3.1 Evaluation settings

For all test and validation sets but *Medline-test2019*, we use *SacreBLEU* with the default settings against untokenized references.[10] When the

[8] Each sentence pair has a probability of 0.8 of being merged with the previous sentence pairs, with a max document size of 1024 tokens or 64 sentences.

[9] Our adapters use near-zero initialization and the original *pre-norm* architecture (Bapna and Firat, 2019), even though our Transformer models are post-norm.

[10] `BLEU+case.mixed+numrefs.1+smooth.exp+tok.13a+version.1.4.3`

test set is document-level, we split it into sentences first, as well as the model outputs and compute regular sentence-level corpus BLEU.

For *Medline-test2019* we use `SacreBLEU` in case insensitive mode with the `intl` tokenization,[11] which mimics closely the evaluation settings of the WMT19 Biomedical task. We use the alignments provided by the organizers, and keep all alignments regardless of their annotation (e.g., `OK` or `NO_ALIGNMENT`) but remove those where one side is marked as "omitted".

3.2 Hyper-parameters

We use the Transformer Big architecture (Vaswani et al., 2017) with post-norm (as prior experiments with pre-norm gave worst results), which we train with fairseq (Ott et al., 2019). We share the source and target embeddings and tie them with the vocabulary projection. We use Adam with warmup and a maximum learning rate of 0.001. Training is done on 4 GPUs with mixed precision and accumulated gradients over 16 updates (Ott et al., 2018). In some cases, we had to reduce the learning rate to 0.0005 because of exploding gradient issues. We use a dropout rate of 0.1 and label smoothing of 0.1. We train for maximum 24 epochs with early stopping according to BLEU on *newsvalid*.

The models using back-translation, BPE-dropout, and/or noise are initialized with the epoch 12 checkpoint of the baseline model and trained for 12 more epochs. The doc-level model is initialized from the sent-level model with *BT + BPE-dropout + noise*, and fine-tuned for 4 more epochs.

The pre-trained MLM is trained with RoBERTa Base's default training settings but uses the same architecture as the NMT encoder (sinusoidal positional embedding, post-norm Transformer). We also remove the non-linear transformation in RoBERTa's LM head. Due to time constraints, we train the MLM for 2 epochs only.

The models initialized with RoBERTa use the RoBERTa Base architecture for the encoder (embedding size of 768 and feed-forward size of 3072), and a Transformer Big with 3 layers only for the decoder. They also use a higher dropout rate of 0.3. As source-side pre-processing, we use the same GPT tokenizer as RoBERTa; and as target-side pre-processing, a monolingual SentencePiece model of size 24k without inline casing.

[11] `BLEU+case.lc+numrefs.1+smooth.exp+tok.intl+version.1.4.3`

3.3 Ensembles

As primary submissions to all three tasks, we use an ensemble of three document-level models. To save computation time, and avoid re-training new models, we ensemble models that were trained with different settings, but whose pre-processing is compatible (see Table 5). To achieve better ensemble results, we train three different instances of Bidirectional Big (for 12 epochs), which serve as initialization for models *8*, *9*, and *10* (fine-tuned for 12 more epochs). These three models are combined with model *7* as ensemble *15*. We continue training these three models with document-level data (for 4 epochs) to create ensemble *19*. Ensembles *18* and *22* are obtained by taking the same models as ensembles *15* and *19*, training domain-specific adapter layers, and combining them again.

3.4 Results

ID	Model	DE-EN	EN-DE
0	FAIR 2019 (single)	41.0	40.9
1	Monodirectional Base	40.7	41.1
2	Bidirectional Base	39.9	40.1
3	Monodirectional Big	**42.0**	41.6
4	Bidirectional Big	41.9	**41.8**

Table 3: Comparison of monodirectional versus bidirectional models. BLEU scores on *newstest2019*. Bidirectional Big serves as a baseline for our next experiments. *FAIR 2019 (single)* is one of the models from the ensemble that ranked first in the WMT19 News Task (Ng et al., 2019).

Baseline models We compare Transformer Base and Transformer Big architectures, and monodirectional (German → English and English → German) versus bidirectional models (German ↔ English). Table 3 shows their results on *newstest2019*.

Robustness to noise Table 4 evaluates the robustness of our models to several forms of synthetic noise and to other types of tokenization.[12] BPE dropout slightly improves robustness to certain types of synthetic noise, and drastically improves robustness to other types of tokenization, especially character-level translation ("spelled out" column). Source-side synthetic noise dramatically

[12] While robustness to tokenization is not necessary for these tasks, it can be a desirable property for an NMT model. For instance, we could reduce the size of the vocabulary for model compression, or change the tokenization algorithm and vocabulary for the model to be compatible with other models (e.g., for ensembling, pre-training, etc.)

ID	Model	Clean	Char noise	No space	No 'e'	Spelled out	Other BPE
0	FAIR 2019 (single)	41.3	15.0	6.2	17.5	–	–
3	Monodirectional Big	42.0	8.4	0.3	10.3	1.4	30.0
5	3 + RoBERTa-12	41.9	11.5	2.4	13.9	–	–
6	3 + RoBERTa-8	**42.4**	10.4	2.0	12.2	–	–
4	Bidirectional Big	42.2	8.9	0.6	10.7	2.1	30.8
7	4 + MLM	41.9	9.1	0.7	10.6	1.6	31.4
8	4 + BT	42.2	9.6	0.7	11.1	1.7	32.1
9	8 + BPE dropout	42.0	10.8	0.8	14.3	22.8	37.5
10	9 + Noise	**42.4**	29.3	8.3	31.2	31.8	38.3
11	10 + Docs	**42.4***	**33.6***	**23.3***	**33.8***	**34.6**	**39.4**

Table 4: Robustness on English-German translation (case-insensitive BLEU) to synthetic noise (random char-level noise, all whitespaces removed or all 'e' letters removed) or to different tokenizations (char-level instead of BPE, or different BPE model than used for training). All the test sets are variants of *newstest2019*. *Char noise* consists in modifying each character with 0.1 probability, with either a deletion, insertion (of an ASCII letter or digit), substitution or swap. *Spelled out* means that we segment the input character-by-character (e.g., `I like pizzas` → `i _ l i k e _ p i z z a s`). With *Other BPE*, we use a different BPE model (trained with SentencePiece on lowercased monolingual news data); and whenever a word piece is out-of-vocabulary, we segment it as characters (i.e., spelling out). Numbers with $\star$ are obtained with document-level translation.

improves robustness to the same type of character-level noise and to the absence of whitespaces or of the 'e' letter.[13] Interestingly, when combined with BPE dropout, it also further improves robustness to other types of tokenization.

Note that the document-level model with noise is even better with noise robustness. This is probably due to its longer training (which means that it has seen more noise).[14] The same model used in sentence-level decoding mode (scores not reported here) achieves similar improvements.

Domain robustness and domain adaptation Table 5 shows the BLEU scores of our models on test sets from multiple domains (News, IT, QED, Medline, Chat). We can assess the domain robustness of our non-adapted models for use in the zero-shot robustness task. It also shows the translation quality of the adapted models on the Biomedical and Chat tasks (on their respective dev sets).

In our case, contrary to what Kudo (2018); Provilkov et al. (2020) observed, subword regularization with BPE dropout brings no clear improvement to BLEU scores on any of the domains.

We see that fine-tuning performs often better than the adapter layers. Yet, because the difference is minor, we settle with adapters for our submissions as they allow us to train one multi-domain model that can be submitted to all three tasks. They also let us participate in the few-shot task with a model that is adapted to a new domain and does not degrade on other domains (which fine-tuning is known to do, because of catastrophic forgetting).

The scores from Tables 3, 4, and 5 are obtained after normalization of our model outputs with Moses' `normalize-punctuation.perl` (Koehn et al., 2007). However, our submissions do not use any punctuation normalization, except for the robustness task (see below).

Task results Table 6 presents the official BLEU results of our primary and contrastive submissions to the three tasks. We always used the same ensemble of document-level models with adapters (*22*) as primary submission, and single document-level model with adapters (*21*) as first contrastive submission. As second contrastive submission, we submitted different models depending on the task (see Table 6's caption): ensemble of RoBERTa-initialized models (*12*), ensemble of sentence-level model (*18*) or *Covid19NMT* model.

3.5 Robustness Task

For this task, we also train a bidirectional Japanese-English model with all the allowed parallel data from the News Task (15.9M lines pairs). We use the same techniques as with German-English: copy

[13] These are examples of perturbations that humans are able to deal with, but NMT models struggle with. For example, try: "Collagus from across th U, and byond, bring valuabl xprinc and skills that strngthn and improv th work of th halth srvic, and bnfit th patints and communitis w srv."

[14] It was trained for 4 more "epochs". But we define an epoch as a fixed number of training examples, which are much longer when we do document-level training ($\approx 5\times$ longer in terms of BPE tokens).

ID	Model	News	IT	QED	Medline	Chat
0	FAIR 2019 (single)	40.9	47.9	24.0	27.0	42.4
3	Monodirectional Big	41.6	48.5	24.6	28.0	39.6
5	3 + RoBERTa-12	41.5	49.7	25.6	27.0	42.2
6	3 + RoBERTa-8	42.0	49.8	25.1	27.3	39.4
12	5 + 6 + Ensemble	43.0	50.5	25.6	27.4	42.6
4	Bidirectional Big	41.8	49.8	24.4	27.5	41.1
7	4 + MLM	41.5	49.1	24.7	27.2	41.3
13	7 + Fine-tuning	–	–	–	30.5	61.3
14	7 + Adapters	–	–	–	30.7	60.4
8	4 + BT	41.8	49.6	25.2	27.4	41.9
9	8 + BPE dropout	41.7	49.8	24.7	27.8	43.3
10	9 + Noise	42.0	49.5	25.1	27.0	41.6
15	7 + 8 + 9 + 10 + Ensemble	43.8	50.9	**25.7**	28.4	43.7
16	10 + Fine-tuning	–	–	–	29.9	61.6
17	10 + Adapters	–	–	–	29.6	61.4
18	7 + 8 + 9 + 10 + Adapt. + Ens.	–	–	–	**31.6**	**62.8**
11	10 + Docs	42.1⋆	49.1	25.2	27.0⋆	44.2⋆
19	8 + 9 + 10 + Docs + Ensemble	**44.3**⋆	**51.0**	25.4	27.8⋆	45.9⋆
20	11 + Fine-tuning	–	–	–	30.3⋆	61.3⋆
21	11 + Adapters	–	–	–	29.9⋆	60.5⋆
22	8 + 9 + 10 + Docs + Adapt. + Ens.	–	–	–	31.2⋆	61.5⋆

Table 5: Domain robustness and domain adaptation on English-German translation (case-sensitive BLEU except for Medline). *News*, *IT*, *QED* and *Medline* are respectively *newstest2019*, *IT-valid*, *QED-valid*, *Medline-test2019* from Table 2. *Chat* is the English-German subset of *BConTrasT-dev*, which contains only the agent's utterances. Numbers with ⋆ are obtained with document-level translation. For *Chat*, we translate the full bilingual dialogues (using both the agent and the customer utterances as context), then compute BLEU on the agent's part only. The models in bold were submitted to one or several tasks (see Table 6).

symbol, inline casing, source-side BPE dropout, and source-side noise. However, we do not train at the document-level, nor do language model pre-training, back-translation, or ensembles. To reduce the effect of the JESC data whose English side is in lowercase, we add source-side corpus tags (Bérard et al., 2019) for all corpora but ParaCrawl (we do not use any corpus tag at test time). We also pre-tokenize the Japanese training data with Kytea, like specified by the organizers.[15]

The test sets for this task are sentence-level. However, we observe that some of the test sets contain lines with several sentences, which causes our models to generate too short outputs. To solve this issue, we sentence-split the test sets (with Moses' `split-sentences.perl` for German and English and basic split for Japanese, which has non-ambiguous end-of-sentence punctuation). Sentences originating from the same line are translated as a document with our document-level models.

[15]With KyTea 0.4.7: `kytea -out tok -model share/kytea/model.bin`

We normalize the punctuation of our model outputs, using `normalize-punctuation.perl` for English, and replacing ASCII double quotes with German-style quotes in German outputs.

Final results are reported in Table 6. The robustness task has two test sets for German-English: *Set 1* (German ↔ English), which appears to be very noisy text extracted from an online forum; and *Set 3* (only German → English), which contains clean and short sentences. The few-shot task lets us use a small corpus (8503 sentence pairs) of the same domain as *Set 3* to try to improve German → English translation quality over *Set 3* while not degrading quality over *Set 1*. We simply take the same models that we submitted to the zero-shot task and train adapters with the German → English in-domain data. Then, when translating *Set 3*, we turn on the adapters and turn them off for *Set 1*.

3.6 Chat Translation Task

For the primary and first contrastive submission, we used our document-level models with chat-domain

Model	Chat		Biomedical		Robustness zero-shot			Few-shot	
					Set 1	Set 1	Set 3	Set 1	Set 3
	EN-DE	DE-EN	EN-DE	DE-EN	EN-DE	DE-EN	DE-EN	DE-EN	DE-EN
Best	**60.4**	**62.0**	**30.4**	**34.8**	**48.0**	**43.9**	**44.7**	?	?
Primary	60.1	61.0	29.6	**34.8**	42.2	43.4	44.0	43.4	**45.4**
Contr. 1	58.8	59.4	28.4	34.3	40.7	42.1	43.4	42.1	44.2
Contr. 2	**60.4**	61.6	**30.4**	34.1*	41.9†	43.5	**44.7**	**43.5**	44.7

Table 6: Results of the three tasks (BLEU scores): top result in each task and scores of our primary and contrastive submissions. We only report results on German $\leftrightarrow$ English. Please refer to the appendix for the results on the other languages. *Primary:* Ensemble of three document-level models with adapters (*22*). *Contrastive 1:* Single document-level model with adapters (*21*). *Contrastive 2:* Ensemble of four sentence-level models with adapters (*18*). †: Ensemble of two RoBERTa-initialized models (*12*). $\star$: *Covid19NMT* model with `<medical>` tag (Bérard et al., 2020). As the Robustness Task organizers did not communicate official results at the time of submission, the numbers reported here are those appearing on the submission website (OCELoT).

adapters (*22* and *21*) to translate the full bilingual dialogues at once. The BLEU scores reported in Table 6 are computed separately for the agent and customer's side of the dialogues. The second contrastive model is bidirectional and sentence-level (*18*), and used to translate the dialogues utterance by utterance (without extra context).

3.7 Biomedical Task

We had issues with document-level decoding output length on the Medline validation and test sets. The number of sentence delimiters in the output does not always match that of the source document, which makes regular BLEU evaluation impossible. We get between 10% and 20% output documents with the wrong length for German-English, and more than 50% for English-German. This length mismatch issue seems to be caused by domain adaptation,[16] as non-adapted models get a perfect length. On the Chat translation task, there is virtually no length mismatch, and up to 10% length mismatch on *newstest2019*, caused by source documents that are close to or above the 1024 tokens limit.

Whenever a length mismatch happens, we revert to sent-level decoding for this particular document. As our English-German submission to the Biomedical task, we used fully sent-level decoding outputs (by our doc-level models), as almost 100% of the document-level outputs had the wrong length.

Our *Covid19NMT* model (Bérard et al., 2020) ranked first in Spanish-English and Italian-English (50.6 and 42.5 BLEU) and lags behind with less than 1 BLEU difference in German-English and French-English (34.1 and 43.1 BLEU).

[16]One likely explanation is that there are some alignment errors in the Medline training data that cause adapted models to ignore the sentence delimiters in some cases. For instance, we observed that the titles are often misaligned (e.g., "INTRODUCTION").

4 Conclusion

We find that, if given enough capacity (e.g., Transformer Big), a single bidirectional model can give similar performance to mono-directional models of the same size.

Like showed by Bapna and Firat (2019), it is possible to perform lightweight domain adaptation using adapter layers, and achieve comparable performance to fine-tuning of the whole model. Thanks to adapter layers added to our bidirectional model, we achieve competitive results on all 3 tasks with one model.

MLM pre-training results for bidirectional models are inconclusive. The pre-trained model seems to be slightly more robust in some aspects, but not as robust to domain shift as one would hope. This may be due to fewer training epochs compared to our previous experiments (Clinchant et al., 2019). RoBERTa pre-training gives promising results in terms of noise robustness; it also seems to bring slight improvements in terms of domain robustness. Note that the models initialized with RoBERTa have fewer parameters than the Transformer Big NMT architecture.

Finally, document-level fine-tuning gives document-level decoding abilities to a bidirectional NMT model without degrading its sentence-level decoding performance. However, document-level decoding does not improve translation quality as measured by BLEU. We also find that generating documents with the right number of sentences (i.e., same length as the input) can be challenging on some test sets.

References

Ankur Bapna and Orhan Firat. 2019. Simple, Scalable Adaptation for Neural Machine Translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1538–1548, Hong Kong, China.

Yonatan Belinkov and Yonatan Bisk. 2018. Synthetic and Natural Noise Both Break Neural Machine Translation. In *International Conference on Learning Representations*.

Alexandre Bérard, Ioan Calapodescu, and Claude Roux. 2019. Naver Labs Europe's Systems for the WMT19 Machine Translation Robustness Task. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 526–532, Florence, Italy.

Alexandre Bérard, Zae Myung Kim, Vassilina Nikoulina, Eunjeong Lucy Park, and Matthias Gallé. 2020. A Multilingual Neural Machine Translation Model for Biomedical Data. In *Proceedings of the EMNLP 2020 Workshop NLP-COVID*, Online.

Isaac Caswell, Ciprian Chelba, and David Grangier. 2019. Tagged Back-Translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 53–63, Florence, Italy.

Stephane Clinchant, Kweon Woo Jung, and Vassilina Nikoulina. 2019. On the use of BERT for Neural Machine Translation. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 108–117, Hong Kong.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual Language Model Pretraining. In *Advances in Neural Information Processing Systems*, pages 7059–7069.

Sergey Edunov, Alexei Baevski, and Michael Auli. 2019. Pre-trained Language Model Representations for Language Generation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4052–4059, Minneapolis, Minnesota.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding Back-Translation at Scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500, Brussels, Belgium.

Markus Freitag and Yaser Al-Onaizan. 2016. Fast Domain Adaptation for Neural Machine Translation.

Marcin Junczys-Dowmunt. 2019. Microsoft Translator at WMT 2019: Towards Large-Scale Document-Level Neural Machine Translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 225–233, Florence, Italy.

Vladimir Karpukhin, Omer Levy, Jacob Eisenstein, and Marjan Ghazvininejad. 2019. Training on Synthetic Noise Improves Robustness to Natural Noise in Machine Translation. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 42–47, Hong Kong, China.

Catherine Kobus, Josep Crego, and Jean Senellart. 2017. Domain Control for Neural Machine Translation. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 372–378, Varna, Bulgaria.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online.

Marco Lui and Timothy Baldwin. 2012. langid.py: An Off-the-shelf Language Identification Tool. In *Proceedings of the ACL 2012 System Demonstrations*, pages 25–30, Jeju Island, Korea.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook FAIR's WMT19 News Translation Task Submission. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 314–319, Florence, Italy.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and

Michael Auli. 2019. fairseq: A Fast, Extensible Toolkit for Sequence Modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota.

Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. 2018. Scaling Neural Machine Translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 1–9, Brussels, Belgium.

Jerin Philip, Alexandre Bérard, Matthias Gallé, and Laurent Besacier. 2020. Monolingual Adapters for Zero-Shot Neural Machine Translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Online.

Ivan Provilkov, Dmitrii Emelianenko, and Elena Voita. 2020. BPE-Dropout: Simple and Effective Subword Regularization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1882–1892, Online.

Sascha Rothe, Shashi Narayan, and Aliaksei Severyn. 2020. Leveraging Pre-trained Checkpoints for Sequence Generation Tasks. *Transactions of the Association for Computational Linguistics*, 8:264–280.

Fahimeh Saleh, Alexandre Bérard, Ioan Calapodescu, and Laurent Besacier. 2019. Naver labs Europe's systems for the document-level generation and translation task at WNGT 2019. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 273–279, Hong Kong.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT Rediscovers the Classical NLP Pipeline.

Jörg Tiedemann. 2012. Parallel Data, Tools and Interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2214–2218, Istanbul, Turkey.

Vaibhav Vaibhav, Sumeet Singh, Craig Stewart, and Graham Neubig. 2019. Improving Robustness of Machine Translation with Synthetic Noise. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1916–1920, Minneapolis, Minnesota.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Elena Voita, Rico Sennrich, and Ivan Titov. 2019. The Bottom-up Evolution of Representations in the Transformer: A Study with Machine Translation and Language Modeling Objectives. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4396–4406, Hong Kong, China.

ID	Model	Clean	Char noise	No space	No 'e'	Spelled out	Other BPE
0	FAIR 2019 (single)	42.6	19.6	13.0	16.1	–	–
3	Monodirectional Big	43.6	14.6	6.8	11.2	2.5	36.6
4	Bidirectional Big	43.5	14.5	8.0	11.9	3.6	35.1
7	4 + MLM	**43.7**	13.9	9.4	10.9	3.8	35.2
8	4 + BT	43.6	14.8	5.5	11.5	4.2	36.5
9	8 + BPE dropout	43.6	17.0	10.2	14.8	26.1	41.6
10	9 + Noise	43.3	33.3	21.8	31.5	34.8	41.6
11	10 + Docs	43.0⋆	**35.2**⋆	**26.3**⋆	**32.9**⋆	**36.1**	**41.8**

Table 7: Robustness on German-English translation (case-insensitive BLEU) to synthetic noise (random char-level noise, all whitespaces removed or all 'e' letters removed) or to different tokenizations (char-level instead of BPE, or different BPE model than used for training). All the test sets are variants of *newstest2019*. Numbers with ⋆ are obtained with document-level translation.

ID	Model	News	IT	QED	Medline	Chat
0	FAIR 2019 (single)	41.0	53.8	34.8	30.0	47.9
3	Monodirectional Big	42.0	**57.8**	35.4	30.6	48.9
4	Bidirectional Big	41.9	57.6	34.4	30.1	47.7
7	4 + MLM	41.9	56.4	34.5	30.0	49.2
13	7 + Fine-tuning	–	–	–	30.9	59.7
14	7 + Adapters	–	–	–	30.9	59.9
8	4 + BT	42.0	56.6	34.8	30.0	48.6
9	8 + BPE dropout	41.9	56.1	35.3	29.7	48.5
10	9 + Noise	41.8	56.0	34.9	29.7	48.2
15	7 + 8 + 9 + 10 + Ensemble	**43.3**	**57.8**	**35.7**	30.5	49.5
16	10 + Fine-tuning	–	–	–	30.8	61.3
17	10 + Adapters	–	–	–	30.2	60.8
18	7 + 8 + 9 + 10 + Adapt. + Ens.	–	–	–	31.4	61.7
11	10 + Docs	41.4⋆	56.0	35.1	30.0⋆	50.5⋆
19	8 + 9 + 10 + Docs + Ensemble	42.8⋆	56.6	**35.7**	30.7⋆	50.7⋆
20	11 + Fine-tuning	–	–	–	30.8⋆	60.9⋆
21	11 + Adapters	–	–	–	31.0⋆	60.5⋆
22	8 + 9 + 10 + Docs + Adapt. + Ens.	–	–	–	**31.7**⋆	**62.1**⋆

Table 8: Domain robustness and domain adaptation on German-English translation (case-sensitive BLEU except for Medline). *News*, *IT*, *QED* and *Medline* are respectively *newstest2019*, *IT-valid*, *QED-valid*, *Medline-test2019* from Table 2. *Chat* is the German-English subset of *BConTrasT-dev*, which contains only the agent's utterances. Numbers with ⋆ are obtained with document-level translation. For *Chat*, we translate the full bilingual dialogues (using both the agent and the customer utterances as context), then compute BLEU on the customer's part only. The models in bold were submitted to one or several tasks (see Table 6).

Task:	Biomedical			Robustness zero-shot			
Pair:	FR-EN	ES-EN	IT-EN	JA-EN		EN-JA	
Set:				1	2	1	2
Best	**44.1**	**50.6**	**42.5**	**26.6**	**15.2**	**37.6**	**29.2**
Ours (primary)	43.1	**50.6**	**42.5**	24.5	13.3	33.3	25.6

Table 9: Results of the Biomedical and Robustness tasks (BLEU scores): top result in each task and scores of our primary submissions. The primary submission to the Biomedical Task in French, Spanish and Italian to English is our multilingual *Covid19NMT* model (Bérard et al., 2020). As the Robustness Task organizers did not communicate official results at the time of submission, the numbers reported here are those appearing on the submission website (OCELoT)

The University of Edinburgh-Uppsala University's Submission to the WMT 2020 Chat Translation Task

Nikita Moghe[1] **Christian Hardmeier**[2] **Rachel Bawden**[1]
[1]School of Informatics, University of Edinburgh, Scotland
[2]Department of Linguistics and Philology, Uppsala University
{nikita.moghe,rachel.bawden}@ed.ac.uk
christian.hardmeier@lingfil.uu.se

Abstract

This paper describes the joint submission of the University of Edinburgh and Uppsala University to the WMT'20 chat translation task for both language directions (English↔German). We use existing state-of-the-art machine translation models trained on news data and fine-tune them on in-domain and pseudo-in-domain web crawled data. We also experiment with (i) adaptation using speaker and domain tags and (ii) using different types and amounts of preceding context. We observe that contrarily to expectations, exploiting context degrades the results (and on analysis the data is not highly contextual). However using domain tags does improve scores according to the automatic evaluation. Our final primary systems use domain tags and are ensembles of 4 models, with noisy channel reranking of outputs. Our en-de system was ranked second in the shared task while our de-en system outperformed all the other systems.[1]

1 Introduction and challenges

The task's aim is to create machine translation (MT) systems to enable task-oriented communication between a service agent and a customer speaking different languages (English and German respectively). Like most dialogues, the texts can show strong context sensitivities, as the customer and the agent engage in a common activity and continually react to each other's utterances (Hardmeier, 2014; Bawden, 2018). However, the dialogues, which relate to ordering or reserving products and services from a limited set of providers, also follow fairly strong scripts and are anchored in a small discourse universe defined by the products on offer. Their context sensitivity is therefore counterbalanced by domain-specific conventions and expectations.

Our design choices are informed by an initial manual inspection of the training data and a baseline translation, which revealed that the main challenges relate to idiomaticity: incorrect or poor translation of English idioms, named entities and politeness markers (e.g. formal vs. informal forms of address, or poor translation of English *sir*) and an incorrect use of domain-specific terminology. Almost always, the problems were the result of an excessively literal translation of the source text, and this literalness also frequently affected the reference translations themselves too. Surprisingly, we found few instances of phenomena explicitly requiring context to be correctly translated (e.g. we did not find pronominal anaphora to be a major problem in the dialogues examined).[2] The context-dependent instances we did find were more task-specific (e.g. English *Enjoy!* should be translated differently depending on whether it is about a pizza (*Guten Appetit!*) or a film (*Viel Spaß!*)).

We therefore focus on domain adaptation and general context modelling strategies. Our submissions are based on existing state-of-the-art MT systems for news translation, which we fine-tune on in-domain and pseudo-in-domain data. We also experiment with (i) adapting the models to the different speaker roles and to the different tasks during fine-tuning and (ii) exploiting preceding context through a simple but effective method of concatenating previous sentences to the current one. Our code and models are publicly available.[3]

[1] http://www.statmt.org/wmt20/chat-task_results_DA.html

[2] We tested AllenNLP's coreference resolution tool (Gardner et al., 2018) on a few examples where pronoun resolution seemed relevant and found that it performed very poorly in these cases, confirming similar conclusions by Bawden (2016). We therefore decided not to model coreference explicitly.

[3] http://github.com/chardmeier/WMT2020-Chat

Proceedings of the 5th Conference on Machine Translation (WMT), pages 473–478
Online, November 19–20, 2020.

2 Data

The task data consists of parallel task-oriented dialogues between an agent (English) and a customer (German) across six domains: (i) ordering pizza, (ii) making auto repair appointments, (iii) ordering a taxi, (iv) ordering movie tickets, (v) ordering coffee and (vi) making restaurant reservations. The dialogues were initially in English, retrieved from a subset of the TaskMaster-1 dataset (Byrne et al., 2019) and then manually translated into German at Unbabel.[4] Although the speaker tags are provided for each utterance, the conversations are not explicitly marked with their task domain. The task being to translate the agent's utterances from English into German and the customer's utterances from German to English, we evaluate each translation direction separately, using only the agent's utterances for en–de translation and the customer's utterances for de–en. For training however, we use the full set of 13,845 utterances for both directions.

3 Approaches

We explore four approaches, each of which is detailed below: (i) pretraining using additional data sources, (ii) speaker adaptation, (iii) domain adaptation and (iv) incorporating previous context.

Pretraining To account for the limited in-domain data, we use pre-existing MT models trained for the WMT'19 news task (Barrault et al., 2019) and then continue training on pseudo-in-domain web crawled data from the Paracrawl project[5] (Bañón et al., 2020), before fine-tuning on the in-domain chat training data. We compare two different base systems for each language direction: UEDIN models[6] ((Bawden et al., 2019a) and FAIR models (Ng et al., 2019). The pseudo-in-domain data on which training is continued is created by filtering Paracrawl data using dual conditional noisy cross-entropy filtering (Junczys-Dowmunt, 2018). This consists in training a neural language model for each language on the task training data, and jointly scoring each parallel sentence in Paracrawl using the two models. We take the top scoring 2.5 million subset of the original 34 million en–de sentences (those that most resemble the task data).

Speaker adaption Distinguishing between the two speaker roles is important as they have different contributions to the dialogue; the customer's utterances are short, interrogative and informal, while the agent's utterances are often long, informative and more formal. We adapt our models to each speaker by using the speaker identity (provided with the task data) as a pseudo-token (Sennrich et al., 2016a): we prepend a speaker tag to each utterance on both the source and the target side.

Domain adaptation Knowing which task the dialogue belongs to (e.g. pizza, film) can be important for disambiguation, as described in Section 1. Similarly to speaker adaptation, we adapt to the different tasks (i.e. domains) by prepending a domain tag to each utterance on both the source and target side. We also consider a setup where all the utterances are tagged with speaker and domain-tags (see the example in Table 1). The dataset consists of chats across six different domains (pizza, auto, taxi, movie, coffee, and restaurant). As the domains are not indicated in the task dataset, we obtain domain tags by automatically classifying each dialogue as belonging to one of the six tasks using the English side of the data and a baseline German translation.

The dialogue classifier is trained by unsupervised k-means clustering of the training set dialogues with scikit-learn (Pedregosa et al., 2011). As features, we use the nouns in the texts (as recognised by the SpaCy PoS tagger[7]), which works substantially better than using all words. The 6 clusters are initialised to the word sets {*pizza*}, {*auto, car, repair*}, {*ride*}, {*movie*}, {*coffee*}, {*dinner, restaurant*}. Dialogues in the test set are then assigned to the cluster with the nearest centroid. To evaluate the classifier, we manually annotated 49 dialogues from the training set. Training only on the remainder of the training data, we achieved perfect accuracy on the annotated set.

To simulate an online translation scenario, we also experimented with classification using only the initial utterances of each dialogue. In this setting, it was beneficial to project the feature space to a very low dimension using Latent Semantic Analysis (LSA). The best results with a macro-averaged F-score of 0.862 (precision 0.896; recall 0.867) were obtained by using the first 4 sentences and an LSA dimensionality of 5. However, since there was no online constraint in the shared task, we ultimately decided to use the more accurate

[4] https://github.com/Unbabel/BConTrasT

[5] https://www.paracrawl.eu

[6] Although the WMT'19 submission included only de–en, we also use the similarly trained model for en–de.

[7] https://spacy.io

Adaptation	Source text	Target text
Speaker	<speaker=customer> Perfect. Okay, got it.	<speaker=customer> Perfekt. In Ordnung, verstanden.
Domain	<taxi> Perfect. Okay, got it.	<taxi> Perfekt. In Ordnung, verstanden.
Speaker+domain	<taxi> <speaker=customer> Perfect. Okay, got it.	<taxi> <speaker=customer> Perfekt. In Ordnung, verstanden.

Table 1: Examples from the dataset annotated with variants of speaker and domain tags.

full-dialogue classifier for our submission.

Context-level MT Finally, we explore using linguistic context (varying numbers of previous utterances) to improve translation, with the aim that previous context can provide vital information for disambiguation or adaptation. We use the approach of concatenating varying numbers of previous sentences to the current sentence, separated by a sentence boundary token `<break>` (Tiedemann and Scherrer, 2017; Bawden et al., 2018). This simple strategy was shown to be one of the most effective in a recent comparison of document-level MT approaches (Lopes et al., 2020). To distinguish between different speakers, we also add the speaker tag to the beginning of every utterance. The models are trained to translate both the context and the utterance into the target language (i.e. n-to-n strategy). The candidate utterance is then extracted from the generated output in a preprocessing step. Since the dialogues are bilingual (the agent and customer are speaking in different languages), the original versions of the previous sentences can be either in English or in German. While we always translate both the context and the current sentence into the target language on the target side, we consider two approaches to incorporate context in the source sentence: (i) `ORIG`: each previous sentence is in the original language of its speaker (if the context and current sentences are not produced by the same speaker, our input will be a mix of English and German) and (ii) `SAME`: the source context is provided in the same language as the current sentence (language consistency in the source input). At test time, this requires translating utterances sentence by sentence (as opposed to batch decoding); when the previous utterances are not from the same speaker, they must first be translated by the MT model in the opposite language direction for them to be used as context for the current sentence.

4 Experimental setup

We compare two neural MT base system types, both WMT'19 news translation task submissions: UEDIN (University of Edinburgh; Bawden et al. 2019a and FAIR (Facebook; Ng et al. 2019). All models are transformer-big models (Vaswani et al., 2017): 6 encoder and 6 decoder layers, model dimension of 1024, 16 heads except that UEDIN has a feedforward dimension of 4096 for both the encoder and decoder, and FAIR models increase this dimension to 8192 in the encoder. UEDIN models are implemented in Marian (Junczys-Dowmunt et al., 2018) and FAIR models in Fairseq (Ott et al., 2019). Both model types are trained on parallel and backtranslated monolingual data from the WMT'19 news translation shared task (Barrault et al., 2019). For our final submission (using the base FAIR model), we also use noisy channel reranking (Yee et al., 2019), which requires MT models in both directions and a (target) language model. We describe the data processing techniques in Appendix A and list the hyper-parameters in Appendix B.

5 Experimental Results and Analysis

We report automatic evaluation results in Section 5.1 and provide a qualitative manual comparison in Section 5.2.

5.1 Automatic evaluation results

We report BLEU scores (Papineni et al., 2002), calculated with SACREBLEU[8] (Post, 2018) on the dev set (beam size of 4).

Pretraining The results in Table 2 show that in-domain fine-tuning of the pretrained models always gives large gains. The pre-trained FAIR models are better than the pre-trained UEDIN models (Barrault et al., 2019). Fine-tuning on filtered paracrawl and then on the in-domain data gives a slight gain for the UEDIN models (particularly for de–en) but slightly degrades the FAIR models. We choose to take as a base the models fine-tuned on filtered paracrawl to fine-tune all subsequent models (with tags and context). Though these models perform similar to the FT_1 models, as these were trained on more data, they are likely to be more robust on unseen data. Note that all pretrained models

[8]Default parameters and case-sensitive evaluation.

outperform the baseline models trained just on the chat training data (shown in the first row).

Model	en–de UEDIN	en–de FAIR	de–en UEDIN	de–en FAIR
Chat baseline	33.2	35.8	37.4	30.9
Pretrained	42.5	41.0	44.9	48.5
+ in-domain (FT_1)	58.6	61.4	61.0	62.3
+ paracrawl (FT_2)	44.8	45.4	46.5	45.2
+ in-domain	58.8	60.8	60.9	62.2

Table 2: BLEU scores on the dev set for both pretrained models, and of each model fine-tuned on (i) in-domain data and (ii) filtered paracrawl then in-domain data.

Effect of adding tags As shown in Table 3, we observe that in general the performance of both systems improves with the addition of tags. The use of speaker tags improves the BLEU scores for UEDIN models while dialogue tags improve the BLEU scores for FAIR models. We did not observe an improvement in BLEU scores in models using both the tags over models that used a single tag.

Model	en–de UEDIN	en–de FAIR	de–en UEDIN	de–en FAIR
FT_2 + no tag	58.8	60.8	60.9	62.2
FT_2 + speaker	59.4	61.3	60.1	62.1
FT_2 + domain	59.6	**61.5**	60.8	**62.7**
FT_2 + speaker + domain	59.6	61.1	61.4	61.6

Table 3: Dev set BLEU scores for fine-tuning with tags.

Context-level MT As shown in Table 4, the contextual models perform similarly to the baseline for FAIR models while the performance degrades slightly with the UEDIN models. Increasing the number of contextual sentences degrades BLEU scores, most likely due to the necessity to translate longer sentences. It is also likely that the MT systems do not benefit from the addition of previous sentences because the particular chat dataset used contains utterances that do not need context to be correctly translated, contrary to expectations but in line with findings by Mosig et al. (2020). Using context in the same language (SAME) was more beneficial than the original context (ORIG). It is evident that SAME would perform better than ORIG as the pre-trained models were never exposed to such mix-language utterances. Despite fine-tuning a monolingual encoder on mix-language utterances, ORIG systems perform well.

Final submission Table 5 shows the results of our primary submission on both the dev and test

Model	en–de UEDIN	en–de FAIR	de–en UEDIN	de–en FAIR
FT_2 + in-domain	58.8	60.8	60.9	62.2
In-domain data uses previous context (ORIG language)				
FT_2 + 1 prev	58.2	60.3	58.9	**61.8**
FT_2 + 2 prev	56.1	60.2	58.7	61.5
FT_2 + 3 prev	53.3	59.5	56.7	61.7
In-domain data uses previous context (SAME language)				
FT_2 + 1 prev	58.1	61.0	59.2	**62.2**
FT_2 + 2 prev	57.5	60.1	59.1	61.5
FT_2 + 3 prev	55.4	60.5	57.3	62.1

Table 4: Dev set BLEU scores for contextual MT models. The numbers before "prev" are the number of previous utterances used as context.

Model(FAIR)	en–de dev	en–de test	de–en dev	de–en test
FT_2 + domain-tags	61.5	**60.3**	62.7	60.6
+ noisy-channel re-ranking	62.0	60.1	62.9	61.8
+ ensemble [primary]	62.1	60.2	63.1	**62.4**
FT_1 [contrastive]	61.4	60.2	62.3	61.8
FT_2 + 1-same [contrastive]	61.0	59.8	62.2	61.5

Table 5: The method-wise ablation of our final submission: a 4-model ensemble of FAIR based FT_2 models fine-tuned with in-domain training data tagged with domain tags. The outputs are obtained through noisy-channel reranking.

sets: a 4-model ensemble, each model trained by first fine-tuning the pre-existing FAIR model on filtered paracrawl data, then on in-domain training data tagged with dialogue tags and then reranked using noisy channel reranking (n=20) (Yee et al., 2019). We note that noisy channel reranking is more effective for en–de than for de–en. Ensembling provides limited gains. We report our contrastive submissions for comparison. Our models were chosen on their respective performances on the dev set. We observe that the trends for dev set and test set are similar except for FT_2 + domain-tags model without the noisy channel re ranking.

5.2 Qualitative Evaluation

As the gains in BLEU scores with different configurations are limited, it is difficult to identify if the models exhibit qualitative improvement. We created an evaluation set by selecting around 40 peculiar utterances in each translation direction from the development set and conducted an informal human evaluation by assigning scores of -1, 0 or 1 to poor, acceptable or particularly good translations. The average score was used to guide model selection. As per the qualitative evaluation, there

were few and similar errors across different models to draw any significant conclusions. Notably, the number of errors was higher for the en–de direction due to the production of literal translations. Our primary submission achieved a score of 85.357 on human evaluation using direct assessment.

6 Discussion and Future Work

We observe that fine-tuning the WMT'19 news-adapted models on in-domain chat data is a strong baseline. The addition of tags, though helpful, has limited gains on BLEU, and the addition of context (intuitively an important component for any dialogue related task) actually degrades results. We speculate that this is due to the nature of the original dataset, which has limited linguistic diversity and utterances that are mostly context-independent (Mosig et al., 2020). The overall translation of this dataset was of excellent quality, allowing easy understanding of the dialogues. However, the translated chats exhibit translationese and in some cases lacked naturalness, also the case of the references themselves. An interesting avenue for data collection would be a spontaneous generation of chats in two different languages which can roughly follow the same discourse as in (Bawden et al., 2019b).

Acknowledgements

We thank Ulrich Germann for providing us with the pretrained UEDIN models and FAIR for making their models publicly available. Christian Hardmeier was supported by the Swedish Research Council under grant 2017-930. This work was supported in part by the UKRI Centre for Doctoral Training in Natural Language Processing, funded by the UKRI (grant EP/S022481/1) and the University of Edinburgh (Moghe). The authors gratefully acknowledge Huawei for their support (Moghe). This work was also supported by funding from the European Union's Horizon 2020 research and innovation programme under grant agreements No 825299 (GoURMET), 825303 and the UK Engineering and Physical Sciences Research Council (EPSRC) fellowship grant EP/S001271/1 (MT-Stretch).

References

Marta Bañón, Pinzhen Chen, Barry Haddow, Kenneth Heafield, Hieu Hoang, Miquel Esplà-Gomis, Mikel L. Forcada, Amir Kamran, Faheem Kirefu, Philipp Koehn, Sergio Ortiz Rojas, Leopoldo Pla Sempere, Gema Ramírez-Sánchez, Elsa Sarrías, Marek Strelec, Brian Thompson, William Waites, Dion Wiggins, and Jaume Zaragoza. 2020. ParaCrawl: Web-scale acquisition of parallel corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4555–4567, Online.

Loïc Barrault, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. 2019. Findings of the 2019 conference on machine translation (WMT19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 1–61, Florence, Italy.

Rachel Bawden. 2016. Cross-lingual pronoun prediction with linguistically informed features. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 564–570, Berlin, Germany.

Rachel Bawden. 2018. *Going beyond the sentence: Contextual Machine Translation of Dialogue*. Ph.D. thesis, LIMSI, CNRS, Université Paris-Sud, Université Paris-Saclay, Orsay, France.

Rachel Bawden, Nikolay Bogoychev, Ulrich Germann, Roman Grundkiewicz, Faheem Kirefu, Antonio Valerio Miceli Barone, and Alexandra Birch. 2019a. The university of Edinburgh's submissions to the WMT19 news translation task. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 103–115, Florence, Italy.

Rachel Bawden, Sophie Rosset, Thomas Lavergne, and Éric Bilinski. 2019b. DiaBLa: A Corpus of Bilingual Spontaneous Written Dialogues for Machine Translation. *CoRR*, abs/1905.13354.

Rachel Bawden, Rico Sennrich, Alexandra Birch, and Barry Haddow. 2018. Evaluating Discourse Phenomena in Neural Machine Translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1304–1313, New Orleans, Louisiana.

Bill Byrne, Karthik Krishnamoorthi, Chinnadhurai Sankar, Arvind Neelakantan, Ben Goodrich, Daniel Duckworth, Semih Yavuz, Amit Dubey, Kyu-Young Kim, and Andy Cedilnik. 2019. Taskmaster-1: Toward a realistic and diverse dialog dataset. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4516–4525, Hong Kong, China.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018.

AllenNLP: A deep semantic natural language processing platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 1–6, Melbourne, Australia.

Christian Hardmeier. 2014. *Discourse in Statistical Machine Translation*. Ph.D. thesis, Uppsala University, Department of Linguistics and Philology, Uppsala, Sweden.

Marcin Junczys-Dowmunt. 2018. Dual conditional cross-entropy filtering of noisy parallel corpora. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 888–895, Belgium, Brussels.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia.

Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations*, ICLR'15, San Diego, California, USA.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium.

António V. Lopes, M. Amin Farajian, Rachel Bawden, Michael Zhang, and André F. T. Martins. 2020. Document-level neural MT: A systematic comparison. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, Online, formerly Lisbon, Portugal.

Johannes E. M. Mosig, Vladimir Vlasov, and Alan Nichol. 2020. Where is the context? – a critique of recent dialogue datasets. *CoRR*, abs/2004.10473.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook FAIR's WMT19 news translation task submission. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 314–319, Florence, Italy.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Controlling politeness in neural machine translation via side constraints. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 35–40, San Diego, California.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany.

Jörg Tiedemann and Yves Scherrer. 2017. Neural machine translation with extended context. In *Proceedings of the Third Workshop on Discourse in Machine Translation*, pages 82–92, Copenhagen, Denmark.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

Kyra Yee, Yann Dauphin, and Michael Auli. 2019. Simple and effective noisy channel modeling for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5696–5701, Hong Kong, China.

JUST System for WMT20 Chat Translation Task

Roweida Mohammed, Mahmoud Al-Ayyoub and Malak Abdullah
Jordan University of Science and Technology
Irbid, Jordan
roweida.221@gmail.com,{maalshbool, mabdullah}@just.edu.jo

Abstract

Machine Translation (MT) is a sub-field of Artificial Intelligence and Natural Language Processing that investigates and studies the ways of automatically translating a text from one language to another. In this paper, we present the details of our submission to the WMT20 Chat Translation Task, which consists of two language directions, English→German and German→English. The major feature of our system is applying a pre-trained BERT embedding with a bidirectional recurrent neural network. Our system ensembles three models, each with different hyperparameters. Despite being trained on a very small corpus, our model produces surprisingly good results.

1 Introduction

The language of chat texts is considered a common language where people are rarely paying attention to correct spelling. Therefore, using the traditional methods of Machine Translation (MT), like dictionaries, is insufficient (Hernández, 2009). As deep learning (DL) models are becoming more evolved and complex, this motivates the natural language processing (NLP) community researchers to employ them for challenging tasks such as MT of informal language, such as what is used in chat. Techniques like contextual word embeddings and pre-trained DL models are becoming very common in natural language generation (NLG) tasks such as MT (Kusner et al., 2015; Zou et al., 2013; Abdullah and Shaikh, 2018; Al-Bdour et al., 2019).

The Chat Translation Task is a new task in the Fifth Conference on Machine Translation (WMT20).[1] Translating chat text, specifically the chats of customer support, is a main and exciting task in the field of MT. This kind of tasks has not been widely considered in previous MT studies, mostly because of the absence of openly existing datasets. The target of this new Chat Translation Task is to translate the customer support chat text from English to German and vice versa. The essential goal of this task is to develop models that can translate conversational text and study the use of multilingual models.

We take part in the WMT20 shared chat translation task in two language directions: English→German and German→English. In this paper, we discuss our submission for this task, which is based on the bidirectional recurrent neural networks (bi-RNN) (Schuster and Paliwal, 1997) and using the pre-trained BERT embedding, known as bert-base-multilingual-cased (Devlin et al., 2018).

This paper is constructed as follows. In Section 2, the task and data descriptions are provided. Section 3 discusses our proposed model. Section 4 shows the experiments we conduct and their results. Finally, the Conclusion is in Section 5.

2 Task and Data Description

The Chat Translation shared task of WMT20 offers participants the opportunity to address a challenging problem faced by many companies today as they expand their customer support units to multiple different languages.

The shared task provides a dataset consisting of a set of conversations between agents and customers. The organizers supplied a corpus for the English-German language pair. Specifically, the task involves translating the chat text of an agent speaking English and a customer speaking German. We are asked to translate the agent's chat text from English to German, and the customer's from German to English.

The dataset used for this shared task depends on the corpus of Taskmaster-1 (Byrne et al., 2019),

[1] http://www.statmt.org/wmt20/chat-task.html

Proceedings of the 5th Conference on Machine Translation (WMT), pages 479–482
Online, November 19–20, 2020.

which has the English language, and it consists of dialogues in six fields. A small part of this dataset was chosen and translated to German. The shared task has been provided with train, development, and test sets in JSON format. Each chat in the data file has a specific structure. Table 1 shows the number of conversations in each file of the dataset.

Dataset	# of Conversation
Train dataset	550
Dev dataset	78
Test dataset	78

Table 1: Number of conversations in each set.

Each conversation contains a speaker (who is either an agent or a customer), a source chat text, and a target chat text. For the test set file, we are asked to translate the source chat text to target depending on the speaker. If it is an agent, the translation is from English to German. Otherwise, the translation is from German to English. For evaluating the participating models, the task organizers employ both automatic metrics (BLEU (Papineni et al., 2002) and TER (Snover et al., 2006)) as well as human evaluation.

3 JUST System

Our System follows the sequence of steps shown in Figure 1. In the following subsections, we discuss each step in details.

3.1 Preprocessing Data

For the dataset preprocessing, we first converted the files from JSON file, as given in the shared task, to text files, so we can work with them easily. The training, dev, and test sets are divided into two groups: one that contains the agent as the speaker (English→German) and one that contains the customer as the speaker (German→English). Table 2 shows the number of examples in each group.

Groups	Train	Dev	Test
Agent	7,629	1,040	1,133
Customer	6,215	862	967

Table 2: Number of examples in each group.

3.2 Extracting Features

After preparing the dataset and preprocessing it, we use the pre-trained BERT model to get the word embeddings of the dataset. Specifically, we use Bert-base-multilingual-cased[2] to extract feature vectors of the dataset to be used in the training of our models. For each word in the sentence of the encoder side, we get a file containing the word's embedding. The same is done for the decoder side.

3.3 The System Architecture

Our system is an adaptation of OpenNMT[3], an open-source toolkit for neural machine translation (NMT) (Klein et al., 2017). It is created on the PyTorch framework (Paszke et al., 2017). After ensuring that the dataset is ready to be trained in our system, we feed our dataset to the bi-RNN with long short-term memory (LSTM) cells (Hochreiter and Schmidhuber, 1997) and an attention mechanism (Luong et al., 2015) along with the word embeddings we extract from the dataset and trained everything jointly. For each different set of hyperparameters, we train the model separately. We save the best three models. Table 3 shows the different hyperparameters used for the three models as well some of the experiments that have been done using GloVe embedding (Pennington et al., 2014) + byte pair encoding (BPE) (Sennrich et al., 2015) with a vocabulary of 10K sub-word units (Experiment-1), GloVe + without BPE (Experiment-2), and the default model. The rest of the hyperparameters are left at their default value.

Models	Batch size	Dropout	BPE	Embedding
Default	64	0.3	Yes	GloVe
Experiment-1	64	0.4	Yes	GloVe
Experiment-2	64	0.3	No	GloVe
Model-A	32	0.6	No	BERT
Model-B	100	0.7	No	BERT
Model-C	182	0.7	No	BERT

Table 3: Different hyper parameters of the three models.

We also experiment with the celebrated Transformer mode (Vaswani et al., 2017). However, this model results in very low BLEU scores when evaluated on the dev set. Moreover, it takes about four days to finish training in one experiment. So, we decide to exclude it from further consideration.

3.4 Model Ensembling

Before the test set is released, we train different models using the training set and evaluate them

[2]https://github.com/google-research/bert
[3]https://opennmt.net/OpenNMT-py/options/train.html

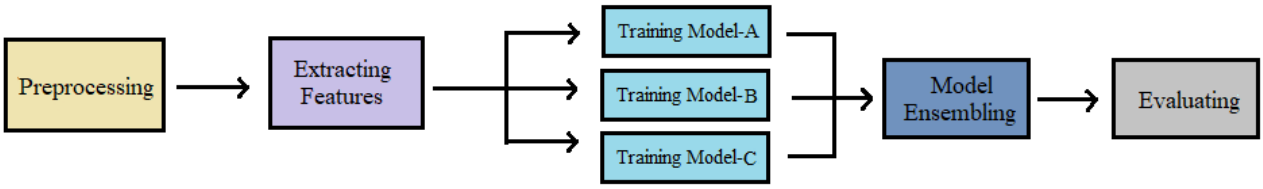


Figure 1: Flowchart of our system.

using the dev set. After training our system, we choose the best three models and ensemble them to get the final output.

4 Results

The results based on the dev set are show in Table 4. The table shows the results of our three models, which we choose for the ensembling step, as well as the other experiments mentioned earlier. The Table shows the difference between them using the BLEU score. From the above table we can notice that training without using BPE improves the results. Moreover, we have chosen the pre-trained BERT because it improves the results compared to the GloVe embedding.

Models	BLEU
Default	32.99
Experiment-1	34.80
Experiment-2	35.21
Model-A	36.88
Model-B	37.07
Model-C	40.93

Table 4: Results of our experiments for the dev dataset.

For evaluation on the test set, we combine the train and dev dataset of each group into one file. Table 5 shows the number of examples in each group after combining them into one file.

	Agent	Customer
Combined train + dev	8,669	7,077

Table 5: Number of examples after combining the files.

We train each group separately and then we ensemble the three models into one. This model is used to get the target of each sentence in the test set of each group. It is worth mentioning that we only use the small dataset provided with the shared task.

Table 6 shows the results for the human evaluation between the human, best score and our model for the English→German scores.

Team	Agent Ave.
Human	91.43
Best	88.21
Our Model	63.93

Table 6: Results of the human evaluation.

Table 7 shows the results we get in the shared task compared to the baseline and the best results. We can see that the agent BLEU score of our model is higher than the baseline, which is translating from English to German. On the other hand, the customer BLEU score for the baseline beat our model, which is translating from German to English.

5 Conclusion

This work describes JUST's submission to the WMT20 chat translation task. For all two translation directions, English→German and German→English, we used the pre-trained BERT embedding with the bi-RNN. We trained one model with different hyperparameters and then ensembled to one final system to translate the test set provided by the shared task. At the end of this work, we find out that a simple NMT model with BERT embedding can achieve surprisingly good results even if it is trained on a very small corpus.

References

Malak Abdullah and Samira Shaikh. 2018. Teamuncc at semeval-2018 task 1: Emotion detection in english and arabic tweets using deep learning. In *Proceedings of the 12th international workshop on semantic evaluation*, pages 350–357.

		Agent BLEU	Customer BLEU	Agent TER	Customer TER
Best	-	60	62	0.25	0.23
FAIR-WMT19	Baseline	43.4	49.7	0.379	0.3195
test1_corpus	Our model	46.4	42.5	0.382	0.4015

Table 7: Results of the shared task.

Ghadeer Al-Bdour, Raffi Al-Qurran, Mahmoud Al-Ayyoub, and Ali Shatnawi. 2019. A detailed comparative study of open source deep learning frameworks. *arXiv preprint arXiv:1903.00102*.

Bill Byrne, Karthik Krishnamoorthi, Chinnadhurai Sankar, Arvind Neelakantan, Daniel Duckworth, Semih Yavuz, Ben Goodrich, Amit Dubey, Andy Cedilnik, and Kyu-Young Kim. 2019. Taskmaster-1: Toward a realistic and diverse dialog dataset. *arXiv preprint arXiv:1909.05358*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Adolfo Hernández. 2009. A ngram-based statistical machine translation approach for text normalization on chat-speak style communications.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Lstm can solve hard long time lag problems. In *Advances in neural information processing systems*, pages 473–479.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M Rush. 2017. Opennmt: Open-source toolkit for neural machine translation. *arXiv preprint arXiv:1701.02810*.

Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. 2015. From word embeddings to document distances. In *International conference on machine learning*, pages 957–966.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11):2673–2681.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of association for machine translation in the Americas*, volume 200. Cambridge, MA.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Will Y Zou, Richard Socher, Daniel Cer, and Christopher D Manning. 2013. Bilingual word embeddings for phrase-based machine translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1393–1398.

Tencent AI Lab Machine Translation Systems for WMT20 Chat Translation Task

Longyue Wang[§] **Zhaopeng Tu**[§] **Xing Wang**[§] **Li Ding**[†*] **Liang Ding**[‡*] **Shuming Shi**[§]
[§]Tencent AI Lab {*vinnylywang,zptu,brightxwang,shumingshi*}*@tencent.com*
[†]Hong Kong Polytechnic University *dingli@oppo.com*
[‡]The University of Sydney *ldin3097@uni.sydney.edu.au*

Abstract

This paper describes the Tencent AI Lab's submission of the WMT 2020 shared task on chat translation in English⇔German. Our neural machine translation (NMT) systems are built on sentence-level, document-level, non-autoregressive (NAT) and pretrained models. We integrate a number of advanced techniques into our systems, including data selection, back/forward translation, larger batch learning, model ensemble, finetuning as well as system combination. Specifically, we proposed a hybrid data selection method to select high-quality and in-domain sentences from out-of-domain data. To better capture the source contexts, we exploit to augment NAT models with evolved cross-attention. Furthermore, we explore to transfer general knowledge from four different pre-training language models to the downstream translation task. In general, we present extensive experimental results for this new translation task. Among all the participants, our German⇒English primary system is ranked the second in terms of BLEU scores.

1 Introduction

Although neural machine translation (NMT, Bahdanau et al., 2015; Vaswani et al., 2017; Gehring et al., 2017) has achieved great progress in recent years, translating conversational text is still a challenging task due to its inherent characteristics such as discourse awareness (Maruf et al., 2018; Wang et al., 2019), informality (Wang et al., 2018; Yang et al., 2019) and personality (Mirkin et al., 2015; Wang et al., 2016). This is a task-oriented chat translation task (Wang et al., 2017a; Farajian et al., 2020), which aims to translating conversations between customers and agents. As a customer and an agent can respectively natively speak in German and English, the systems should translate the customer's utterances in German⇒English (De⇒En) while the agent's in German⇐English (De⇐En).

In this paper, we present our submission to the novel task in De⇔En. We explore a breadth of established techniques for building Chat NMT systems. Specifically, our systems are based on the self-attention networks including both sentence- and document-level Transformer (Vaswani et al., 2017; Wang et al., 2017b). Besides, we investigated non-autoregressive translation (NAT) models augmented with our recently proposed evolved cross-attention (Ding et al., 2020). Technically, we used the most recent effective strategies including back/forward translation, data selection, domain adaptation, batch learning, finetuning, model ensemble and system combination. Particularly, we proposed a multi-feature data selection on large general-domain data. We not only use three language models (i.e. n-gram, Transformer and BERT based LMs) to filter low-quality sentences, but also employ feature decay algorithms (FDA, Biçici and Yuret, 2011) to select domain-relevant data. In addition, we explore large batching (Ott et al., 2018) for this task and found that it can significantly outperform models with regular batching settings. To alleviate the low-resource problem, we employ large scale pre-training language models including monolingual BERT (Devlin et al., 2019a), bilingual XLM (Conneau and Lample, 2019) and multilingual mBART (Liu et al., 2020), of which knowledge can be transferred to chat translation models.[1] For better finetuning, we investigate homogenous and heterogeneous strategies (e.g. from sentence-level to document-level architectures). Simultaneously, we conduct fully-adapted data processing, model ensemble, back/forward translation and system combination.

[*] This work was conducted when Li Ding and Liang Ding were interning at Tencent AI Lab. Li Ding is now working at OPPO Research Institute.

[1] We experimented mBART after the official submission.

Proceedings of the 5th Conference on Machine Translation (WMT), pages 483–491
Online, November 19–20, 2020.

According to the official evaluation results, our systems in De⇒En and De⇐En are respectively ranked 2nd and 4th.[2] Furthermore, a number of advanced technologies reported in this paper are also adapted to our systems for biomedical translation (Wang et al., 2020) and news translation (Wu et al., 2020) tasks, which respectively achieve up to 1st and 2nd ranks in terms of BLEU scores. Though our empirical experiments, we gain some interesting findings on the chat translation task:

1. The presented data selection method improves the baseline model by up to +18.5 BLEU points. It helps a lot for small-scale data.
2. The large batch learning works well, which makes sentence-level NMT models perform the best among different NMT models.
3. Our proposed method can improve the NAT model by +0.6 BLEU point, which is still hard to beat its autoregressive teachers.
4. Document-level contexts are not useful on the chat translation task due to the limitation of contextual data.
5. It is difficult to transfer general knowledge from pretrained LMs to the downstream translation task.

The rest of this paper is organized as follows. Section 2 introduces data statistics and our processing methods. In Section 3, we present our system with four different models: sentence-level NMT, document-level NMT, non-autoregressive NMT and NMT with pre-training LMs. Section 4 describes advanced technique integrated into our systems such as data selection and system combination. In Section 5, we reports ablation study and experimental results, which is followed by our conclusion in Section 6.

2 Data and Processing

2.1 Data

The parallel data we use to train NMT systems consist of two parts: in-domain and out-of-domain corpora. The monolingual data used for back/forward translation are all out-of-domain. Table 1 shows the statistics of data in En-De.

[2]The primary systems are ranked according to BLEU. And the official results are listed in http://www.statmt.org/wmt20/chat-task_results_DA.html.

Data	# Sents	# Ave. Len.
Parallel		
In-domain	13,845	10.3/10.1
Valid	1,902	10.3/10.2
Test	2,100	10.1/10.0
Out-of-domain	46,074,573	23.4/22.4
+filter	33,293,382	24.3/23.6
+select	1,000,000	21.4/20.9
Monolingual		
Out-of-domain De	58,044,806	28.0
+filter	56,508,715	27.1
+select	1,000,000	24.2
Out-of-domain En	34,209,709	17.2
+filter	32,823,301	16.6
+select	1,000,000	14.5

Table 1: Data statistics after pre-processing. Note that in-domain/valid/test set is speaker-ignored combined and their average lengths are counted based on En/De.

In-domain Parallel Data The small-scale in-domain corpus is constructed by the task organizer.[3] The training, validation and test sets contain utterances in task-based dialogues with contextual information. We use both w/ and w/o context formats for training corresponding models. Although there exists duplicated/noisy sentences, we do not further filter such limited data.

Out-of-domain Parallel Data The participants are allowed to use all the training data in the News shared task.[4] Thus, we combine six corpora including Europarl, ParaCrawl, CommonCrawl, TildeRapid, NewsCommentary and WikiMatrix. We first filter noisy sentence pairs (as detailed in Section 2.2) and simultaneously select parts of them as pseudo-in-domain data (as detailed in Section 4.1).

Out-of-domain Monolingual Data Due to the high degree of sentence similarity within the TaskMaster monolingual corpus,[5] participants are not allowed to use the in-domain monolingual data to train their systems. Thus, we collect part of monolingual data in news domain, which consists of CommonCrawl and NewsCommentary. We conduct data selection (in Section 4.1) to select similar amount of sentences for back/forward translation.

[3]https://github.com/Unbabel/BConTrasT.

[4]http://www.statmt.org/wmt20/translation-task.html.

[5]https://github.com/google-research-datasets/Taskmaster.

We do not use larger monolingual corpora (e.g. CommonCrawl) and leave this for future work.

2.2 Processing

Pre-processing To pre-process the raw data, we employ a series of open-source/in-house scripts, including full-/half-width conversion, Unicode conversation, punctuation normalization, tokenization and true-casing. After filtering steps, we generate subwords via Joint BPE (Sennrich et al., 2016b) with 32K merge operations.

Filtering To improve the quality of data, we filter noisy sentence pairs according to their characteristics in terms of language identification, duplication, length, invalid string and edit distance. According to our observations, the filtering method can significantly reduce noise issues including misalignment, translation error, illegal characters, over-translation and under-translation.

Post-processing After decoding, we process detokenizer and de-truecaser on system outputs. We found that the toolkit can not precisely deal with all cases. Thus, we automatically fix these bugs according to bilingual agreement.

3 Models

We adopt four different model architectures namely: SENT, DOC, NAT and PRETRAIN.

3.1 Sentence-level NMT (SENT)

We use standard TRANSFORMER models (Vaswani et al., 2017) with two customized settings. Due to data limitation, we use the small settings (SENT-S)[6] with regular batch size (4096 tokens $\times$ 8 GPUs). Based on the base settings (SENT-B),[7] we also empirically adopt big batch learning (Ott et al., 2018) (16348 tokens $\times$ 4 GPUs) with larger dropout (0.3).

3.2 Document-level NMT (DOC)

To improve discourse properties for chat translation, we re-implement our document-level model (Wang et al., 2017b) on top of TRANSFORMER. Its addition encoder reads $N = 3$ previous source sentences as history context and the representations are integrated into the standard NMT

[6] https://github.com/pytorch/fairseq/blob/master/fairseq/models/transformer.py#L947.

[7] https://github.com/pytorch/fairseq/blob/master/fairseq/models/transformer.py#L902.

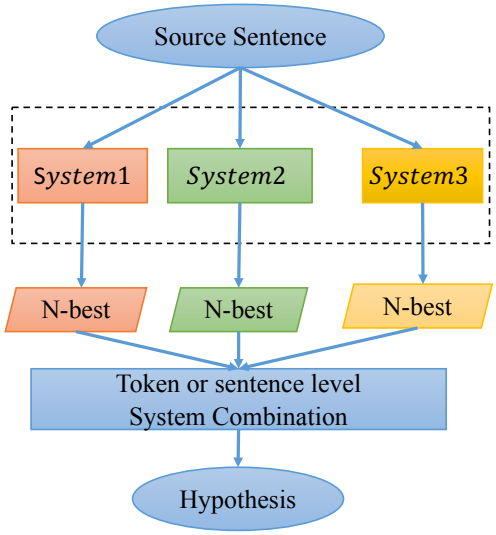


Figure 1: The simplified system combination process, into which we feed each system/model with the source sentence, in turn obtain corresponding n-best result. After pooling all system results, we can perform the token-level or sentence-level system combination decoding and obtain the final hypothesis.

for translating the current sentence. The other configures are same as SENT with small settings.

3.3 Non-autoregressive NMT (NAT)

Different from autoregressive NMT models that generate each target word conditioned on previously generated ones, NAT models break the autoregressive factorization and produce target words in parallel (Gu et al., 2018). Although NAT is proposed to speed up the inference, we exploit it to alleviate sequential error accumulation and improve the diversity in conversational translation. To adequately capture the source contexts, we proposed evolved cross-attention for NAT decoder by modeling the local and global attention simultaneously (Ding et al., 2020). Accordingly, we implement our method based on the advanced MaskPredict model (Ghazvininejad et al., 2019)[8], which uses the conditional mask LM (Devlin et al., 2019a) to iteratively generate the target sequence from the masked input.

3.4 Pretraining NMT (PRETRAIN)

To transfer the general knowledge to chat translation models, we explore to initialize (part of) model parameters with different pretrained language/generation models. Li et al. (2019) showed

[8] https://github.com/facebookresearch/Mask-Predict.

#CP	En-De	De-En
1	60.32	59.51
5	**60.33**	**59.53**
10	60.26	59.42
15	60.19	59.34
20	60.23	59.22
ENS	**60.49**	**60.08**

(a) Model average and ensemble.

#BM	En-De	De-En
4	60.33	59.23
8	60.33	**59.53**
12	60.33	59.24
14	60.34	59.27
16	**60.37**	59.28
20	60.28	59.19

(b) Beam size.

#LP	En-De	De-En
0.8	57.78	57.27
0.9	57.82	57.31
1.0	57.83	57.46
1.1	**57.90**	**57.50**
1.2	57.84	57.49
1.3	57.82	57.49

(c) Length penalty.

Table 2: Effects of different hyper-parameters on translation quality of SENT-B model. The BLEU score is calculated based on *combined* and *tokenized* validation set by *muti-bleu.perl*, which is different from official evaluation.

that large scale generative pretraining could be used to initialize the the document-level translation model by concatenating the current sentence and its context. We follow their work to build the BERT→DOC model. Furthermore, Conneau and Lample (2019) proposed to directly train a novel cross-lingual pretraining language model (XLM) to facilitate translation task. Accordingly, we adopt XLM pretrained model[9] to sentence-level NMT (XLM→SENT). More recently, Liu et al. (2020) proposed a sequence-to-sequence denoising auto-encoder pre-trained on large-scale monolingual corpora in many languages using the BART objective. As they showed promising results on document translation, we additionally conducted the experiment on Chat data after submitting our systems.[10]

4 Approaches

We integrated advanced techniques into our systems, including data selection, model ensemble, back/forward translation, larger batch learning, finetuning, and system combination.

4.1 Data Selection

Inspired by Ding and Tao (2019), multi-feature language modelling can select high-quality data from a large monolingual or bilingual corpus. We present a four-feature selection criterion, which scoring each sentence by BERT LM (Devlin et al., 2019b), Transformer LM (Bei et al., 2018), N-gram LM (Stolcke, 2002) and FDA (Biçici and Yuret, 2011). Three LMs are complement each other on measuring qualities of sentences while FDA can measure its domain relevance given a in-domain dataset. Sentence pairs in the out-of-domain corpus are ranked by a sum of the above feature scores, and we selected top-M instances as pseudo-in-domain data. According to our observations, the selected data can maintain both high-quality and in-domain properties. For BERT LMs, we exploit two models built by Google[11] and our Tencent AI Lab, which are trained on massive multilingual data. The Transformer LM is trained on all in-domain and out-of-domain data via Marian.[12] Besides, we used FDA toolkit[13] to score domain relevance between in-domain and out-of-domain data.

4.2 Checkpoint Average and Model Ensemble

For each model, we stored the top-L checkpoints according to their BLEU scores (instead of PPL or training time) on validation set and generated a final checkpoint with averaged weights to avoid stochasticity. To combine different models (maybe different architectures), we further ensembled the averaged checkpoints in each model. In our preliminary experiments, we find that this hybrid combination method outperforms solely combining checkpoints or models in terms of robustness and effectiveness.

4.3 Finetuning

We employ various finetuning strategies at different phases of training. For Sent-Out→Sent-In finetune (same architecture but different data), we first train a sentence-level model on large pseudo-in-domain data and then continuously train it on small in-domain data. We apply similar strategy for Doc-Out→Doc-In finetuning, and the only difference is to use document-level data. However, pseudo-in-domain data have no document-level contexts

[9] https://github.com/facebookresearch/XLM.

[10] https://github.com/pytorch/fairseq/tree/master/examples/mbart.

[11] https://github.com/google-research/bert.

[12] https://github.com/marian-nmt/marian.

[13] https://github.com/bicici/FDA.

Method	# Sent.	BLEU
SENT-B	10K	41.87
+Bi-FDA	300K	59.36
	500K	59.81
	1M	**59.96**
+Bi-FDA-XL	500K	59.86
	800K	**59.95**
	1M	59.68
+Mono-FDA-XL	800K	**60.36**
	1M	59.80

Table 3: BLEU scores of SENT-BASE model on En⇒De task with different FDA variants (three LMs scoring are consistent).

and we use "$\langle/s\rangle$" symbols as their pseudo contexts (Kim et al., 2019; Li et al., 2020). Besides, we conduct Sent-Out→Doc-In finetuning (different architectures and data). Specifically, we first train a sentence-level model on pseudo-in-domain data and then use parts of corresponding parameters to warm-up a document-level model, which will be continuously trained on in-domain data.

4.4 Back/Forward Translation

Following Section 2, we obtain processed monolingual data. For back translation (BT), we use the best backward translation model to translate from target to source side and produce the synthetic corpus, which is used to enhance the autoregressive NMT models (Sennrich et al., 2016a). About forward translation (FT), we employ forward translation model to perform sequence distillation for NAT models (Kim and Rush, 2016) .

4.5 System Combination

As shown in Figure 1, in order to take full advantages of different systems ($\mathcal{M}odel_1$, $\mathcal{M}odel_2$ and $\mathcal{M}odel_3$), we explore both token- and sentence-level combination strategies.

Token-level We perform token-level combination with confusion network. Concretely, our method follows Consensus Network Minimum Bayes Risk (ConMBR) (Sim et al., 2007), which can be modeled as $E_{ConMBR} = \text{argmin}_{E'}\mathcal{L}(E', E_{con})$, where E_{con} was obtained as backbone through performing consensus network decoding.

Sentence-level We employ the reranking strategy to combine sentence-level systems. Particularly, the sentence reranker contains the best left-to-right (L2R) translation model, R2L (right-to-left) translation model and T2S (target-to-source) translation model. They are integrated by K-best batch MIRA training (Cherry and Foster, 2012) on valid set.

Systems	Integration	BLEU
Models		
SENT-B	IN	42.56
	IN+OUT	**59.81**
SENT-S	IN	41.87
	IN+OUT	58.62
DOC	IN	45.65
	IN+OUT	51.12
	IN→IN	51.93
NAT	IN+OUT	54.01
	*IN+OUT	54.59
Pretrain		
SENT→DOC	OUT→IN	49.77
	OUT→IN+OUT	51.58
XLM→SENT	IN+OUT	**59.61**
BERT→DOC	IN+OUT	56.01
MBART→SENT	IN+OUT	57.48

Table 4: BLEU scores of SENT, DOC, NAT and PRE-TRAIN with different finetuning strategies on En⇒De.

5 Experimental Results

Unless otherwise specified, reported BLEU scores are calculated based on *combined* and *tokenized* validation set by *muti-bleu.perl*, which is different from the official evaluation method.

5.1 Ablation Study

Table 2 investigates effects of different settings on translation quality. We then apply the best hyper-parameters to the models in Section 4 if applicable.

Effects of Model Average and Ensemble Following Section 4.2, we averaged top-L checkpoints in SENT-B model and found that it performs best when $L = 5$. We followed the same operation for SENT-S model and then combined two best averaged models (one from SENT-B and the other one from SENT-S) via ensemble method. As shown in Table 2(a), the ENS model (i.e. "average + ensemble") performs the best.

Effects of Beam Size and Length Penalty Table 2(b) and 2(c) report BLEU scores of SENT-B model using different beam size and length penalty,

# Methods	En⇒De	De⇒En
SENT-S	**59.12**	**59.61**
+BT	59.05	59.22
SENT-B	**60.33**	**59.53**
+BT	59.34	58.94
+FT	59.80	58.94
NAT	54.01	56.58
+FT	**56.56**	**56.69**
XLM	**59.61**	**60.96**
+BT	59.43	58.84

Table 5: BLEU scores of back-translation and forward-translation strategies for different models.

Models	En⇒De		De⇒En	
	-Dom.	+Dom.	-Dom.	+Dom.
Valid Set (combined)				
SENT-S	**60.47**	60.31	**62.66**	61.19
SENT-B	**62.28**	62.08	**64.99**	63.00
XLM	**61.12**	60.85	64.19	**61.30**
Valid Set (split)				
SENT-S	**60.69**	60.48	60.05	**62.09**
SENT-B	61.65	**61.93**	59.64	**63.31**
XLM	**60.90**	60.74	61.12	**62.04**
AVE.	**61.08**	61.05	62.27	**62.48**

Table 6: BLEU scores of domain adaptation strategy for different models.

respectively. As seen, it obtains the best performance when using larger beam size (e.g. 8 or 16). The length penalty prefers around 1.0 because En and De belong to similar language family.

5.2 Main Results

This section mainly reports translation qualities across different models and approaches (in Section 3 and 4). Finally we combine all of them via techniques integration and system combination.

Data Selection Table 3 demonstrates the translation performances of SENT-BASE on different FDA variants. "+Bi-FDA" means using bilingual in-domain data as seed to select N sentences from out-of-domain data while "+Bi-FDA-XL" indicates using larger seed (iteratively add selected pseudo-in-domain data to seed). "Mono" means that we only use source-side data for data selection. As seen, selected data from News domain can help to significantly improve translation quality. However, monolingual selection ("+Mono-FDA-XL") performs better than bilingual one ("+Bi-FDA-XL") and obtain the best BLEU score when $N = 800K$.

Models and Pretraining Table 4 illustrates the translation performances of various NMT models (i.e. SENT, DOC, NAT) with different training strategies. As seen, all models are hungry for larger in-domain data due to the data limitation problem (IN+OUT vs. IN). About sentence-level models, the "base + big batch" setting performs better than the "small" one (SENT-B vs. SENT-S). However, it is difficult for document-level models to outperform sentence-level ones (DOC vs. SENT). The interesting finding is that the document-level model trained on pseudo contexts ("IN+OUT") can improve the baseline that is trained on only real context ("IN") by +5.47 BLEU points. We think there are two main reasons: 1) it lacks of large-scale training data with contextual information; 2) it is still unclear how the context help document translation, which is similar to the conclusion in previous work(Kim et al., 2019; Li et al., 2020). About NAT models, our proposed approach can improve the vanilla NAT by +0.6 BLEU point, which are lower than those of autoregressive NMT models.

About pre-training, we first explore SENT→DOC, which train a sentence-level model and then use part of their parameters to warm-up a document-level model. However, it is still lower than sentence-level models. The performance of BERT→DOC is much better than pure document-level models (56.01 vs. 51.93), which confirms our hypothesis that contextual data is limited in this task. Furthermore, the XLM→SENT can obtain 59.61 BLEU points which are closed to that of SENT-B. The MBART→SENT with CC25 pretrained model can achieve 57.48 BLEU points. We find that performances of most pretraining models can not beat that of the best sentence-level model. There are two possible reasons: 1) needing a number of tricks on finetuning; 2) it is not easy to transfer general knowledge to downstream specific tasks.

Back/Forward Translation Table 5 empirically shows the translation performances of BT and FT for different models, including SENT-S, SENT-B, NAT and PRE-TRAIN. In particular, we performed BT for all systems except NAT, while deploying FT on NAT and SENT-B. As seen, augmenting with monolingual data via BT/FT can not achieve better performances than pure models. The reason

Combination type	En⇒De	De⇒En
Token-level	58.91	59.53
Sentence-level	60.41	62.41

Table 7: Model performance after system combination.

may be that we only use a small part of large-scale monolingual data in news domain. In future work, we will exploit to select in-domain data from the larger monolingual corpus.

Sub-domain Adaptation Modeling of all the speakers and language directions involved in the conversation, where each can be regarded as a different sub-domain. We conduct domain adaptation for different models to avoid performance corruption caused by domain shifting in Table 6. Specifically, we finetune the well-trained models w/ and w/o domain adaptation, denoted as "-Dom." and "+Dom.", and evaluated them on domain *combined* and *split* valid sets. As seen, domain adaptation helps De⇒En more on valid set ("AVE." 61.27 vs. 61.48), while has no much benefits on En⇒De tasks. While evaluating on combined valid sets has a bias towards models without domain adaptation. We attribute this interesting phenomenon to personality and will explore it in the future.

System Combination In order to make full use of the optimal models obtained by the above strategies, we perform token- and sentence-level system combination simultaneously. For each strategy, we generate the n-best candidates to perform the combination. As shown in Table 7, although token-level combination preserves more lexical diversity and avoids the stochasticity, its translation performance is significantly weaker (averagely -2.19 BLEU points) than sentence-level combination. Encouragingly, the sentence-level combination outperforms token-level one on valid set, which is thus used in our final system (in Table 8).

5.3 Official Results

The official automatic evaluation results of our submissions for WMT 2020 are presented in Table 8. For the primary submission, the SYS-1 combines SENT (ensembled SENT-B and SENT-S), DOC and NAT models. As contrastive submissions, the SYS-2 combines SENT and XLM models while the SYS-3 combines SENT, DOC, NAT and XLM ones. Among participated teams, our primary systems achieve the second and the forth BLEU scores on De⇒En and En⇒De, respectively.

Systems	En⇒De		De⇒En	
	Valid	Test	Valid	Test
SYS-1	60.41	58.6	62.41	62.3
SYS-2	58.91	53.6	59.53	54.0
SYS-3	60.42	58.6	62.40	61.9
BEST	–	60.4	–	62.4

Table 8: Official BLEU scores of our submissions for WMT20 Chat task. The BEST denotes the best BLEU scores of systems submitted by participants.

6 Conclusion

The paper is a system description for the Tencent AI Lab's entry into the WMT2020 Chat Translation Task. We explore a breadth of established techniques for building chat translation systems. The paper includes numerous models making use of sentence-level, document-level, non-autoregressive NMT. It also investigates a number of advanced techniques including data selection, model ensemble, finetuing, back/forward translation and initialization using a pretrained LMs. We present extensive experimental results and hope that this work could help both MT researchers and industries to boost the performance of discourse-aware MT systems (Hardmeier, 2014; Wang, 2019).

Acknowledgments

The authors wish to thank the organizers of WMT2020 Chat Translation for their prompt responses on our questions. The authors also specially thank Dr. Xuebo Liu (University of Macau) and Dr. Siyou Liu (Macao Polytechnic Institute), who kindly support us by their engineering and linguistic suggestions, respectively.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR*.

Chao Bei, Hao Zong, Yiming Wang, Baoyong Fan, Shiqi Li, and Conghu Yuan. 2018. An empirical study of machine translation for the shared task of WMT18. In *WMT*.

Ergun Biçici and Deniz Yuret. 2011. Instance selection for machine translation using feature decay algorithms. In *WMT*.

Colin Cherry and George Foster. 2012. Batch tuning strategies for statistical machine translation. In *NAACL*.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *NeurIPS*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019a. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019b. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

Liang Ding and Dacheng Tao. 2019. The university of sydney's machine translation system for wmt19. In *WMT*.

Liang Ding, Longyue Wang, Di Wu, Dacheng Tao, and Tu Zhaopeng. 2020. Localness matters: The evolved cross-attention for non-autoregressive translation. In *COLING*.

M. Amin Farajian, António V. Lopes, André F. T. Martins, Sameen Maruf, and Gholamreza Haffari. 2020. Findings of the wmt 2020 shared task on chat translation. In *Proceedings of the Fifth Conference on Machine Translation*.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. In *ICML*.

Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. 2019. Mask-predict: Parallel decoding of conditional masked language models. In *EMNLP*.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor OK Li, and Richard Socher. 2018. Non-autoregressive neural machine translation. In *ICLR*.

Christian Hardmeier. 2014. *Discourse in statistical machine translation*. Ph.D. thesis, Acta Universitatis Upsaliensis.

Yoon Kim and Alexander M Rush. 2016. Sequence-level knowledge distillation. In *EMNLP*.

Yunsu Kim, Duc Thanh Tran, and Hermann Ney. 2019. When and why is document-level context useful in neural machine translation? In *DiscoMT*.

Bei Li, Hui Liu, Ziyang Wang, Yufan Jiang, Tong Xiao, Jingbo Zhu, Tongran Liu, and Changliang Li. 2020. Does multi-encoder help? a case study on context-aware neural machine translation. *arXiv preprint arXiv:2005.03393*.

Liangyou Li, Xin Jiang, and Qun Liu. 2019. Pretrained language models for document-level neural machine translation. *arXiv*.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *arXiv preprint arXiv:2001.08210*.

Sameen Maruf, André FT Martins, and Gholamreza Haffari. 2018. Contextual neural model for translating bilingual multi-speaker conversations. In *Proceedings of the Third Conference on Machine Translation: Research Papers*.

Shachar Mirkin, Scott Nowson, Caroline Brun, and Julien Perez. 2015. Motivating personality-aware machine translation. In *EMNLP*.

Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. 2018. Scaling neural machine translation. In *WMT*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data. In *ACL*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *ACL*.

Khe Chai Sim, William J Byrne, Mark JF Gales, Hichem Sahbi, and Philip C Woodland. 2007. Consensus network decoding for statistical machine translation system combination. In *ICASSP*.

Andreas Stolcke. 2002. Srilm-an extensible language modeling toolkit. In *ICSLP*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NeurIPS*.

Longyue Wang. 2019. *Discourse-aware neural machine translation*. Ph.D. thesis, Dublin City University.

Longyue Wang, Jinhua Du, Liangyou Li, Zhaopeng Tu, Andy Way, and Qun Liu. 2017a. Semantics-enhanced task-oriented dialogue translation: A case study on hotel booking. In *IJCNLP*.

Longyue Wang, Zhaopeng Tu, Shuming Shi, Tong Zhang, Yvette Graham, and Qun Liu. 2018. Translating pro-drop languages with reconstruction models. In *AAAI*.

Longyue Wang, Zhaopeng Tu, Xing Wang, and Shuming Shi. 2019. One model to learn both: Zero pronoun prediction and translation. In *EMNLP*.

Longyue Wang, Zhaopeng Tu, Andy Way, and Qun Liu. 2017b. Exploiting cross-sentence context for neural machine translation. In *EMNLP*.

Longyue Wang, Xiaojun Zhang, Zhaopeng Tu, Andy Way, and Qun Liu. 2016. The automatic construction of discourse corpus for dialogue translation. In *LREC*.

Xing Wang, Zhaopeng Tu, Longyue Wang, and Shuming Shi. 2020. Tencent AI Lab machine translation systems for the WMT20 biomedical translation task. In *Proceedings of the Fifth Conference on Machine Translation*.

Shuangzhi Wu, Xing Wang, Longyue Wang, Fangxu Liu, Jun Xie, Zhaopeng Tu, Shuming Shi, and Mu Li. 2020. Tencent neural machine translation systems for the WMT20 news translation task. In *Proceedings of the Fifth Conference on Machine Translation*.

Jingxuan Yang, Jianzhuo Tong, Si Li, Sheng Gao, Jun Guo, and Nianwen Xue. 2019. Recovering dropped pronouns in Chinese conversations via modeling their referents. In *NAACL*.

Combining Sequence Distillation and Transfer Learning for Efficient Low-Resource Neural Machine Translation Models

Raj Dabre **Atsushi Fujita**
National Institute of Information and Communications Technology
3-5 Hikaridai, Seika-cho, Soraku-gun, Kyoto, 619-0289, Japan
firstname.lastname@nict.go.jp

Abstract

In neural machine translation (NMT), sequence distillation (SD) through creation of distilled corpora leads to efficient (compact and fast) models. However, its effectiveness in extremely low-resource (ELR) settings has not been well-studied. On the other hand, transfer learning (TL) by leveraging larger helping corpora greatly improves translation quality in general. This paper investigates a combination of SD and TL for training efficient NMT models for ELR settings, where we utilize TL with helping corpora twice: once for distilling the ELR corpora and then during compact model training. We experimented with two ELR settings: Vietnamese–English and Hindi–English from the Asian Language Treebank dataset with 18k training sentence pairs. Using the compact models with 40% smaller parameters trained on the distilled ELR corpora, greedy search achieved 3.6 BLEU points improvement in average while reducing 40% of decoding time. We also confirmed that using both the distilled ELR and helping corpora in the second round of TL further improves translation quality. Our work highlights the importance of stage-wise application of SD and TL for efficient NMT modeling for ELR settings.

1 Introduction

Neural machine translation (NMT) (Bahdanau et al., 2015; Sutskever et al., 2014) enables end-to-end training of translation models and is known to give state-of-the-art results for a large variety of language pairs. NMT models with large hidden sizes or deep stacked layers tend to give better translations than those with small hidden sizes or fewer layers. Large models inevitably need more storage space and computation, and are difficult to deploy on low-computation and low-memory devices. Additionally, beam search decoding is known to improve translation quality but needs more computation and is unacceptable in a low-latency real-time application where faster decoding is as valuable as if not more valuable than translation quality. Consequently, neural models that are compact and fast are extremely important and a growing body of research known as neural model efficiency focuses on this issue.

One of the most popular techniques to train efficient models is knowledge distillation (Hinton et al., 2015) which relies on transferring the knowledge learned by a large model (called teacher) into a smaller model (called student). Sequence distillation (SD) (Kim and Rush, 2016) is a special case of knowledge distillation for sequence-to-sequence models, such as those used for NMT. Not only does it help in the training of compact and fast models with high translation quality, it sometimes helps in eliminating the need for beam search which further increases decoding speed. SD relies on the creation of distilled parallel corpora by translating the training source sentences into the target language by using a large model. The distilled corpora are simplified representations of how the large model sees the original corpora and their quality will have a direct impact on the translation quality of compact models trained with them.

While SD is known to perform extremely well for high-resource settings, its direct application to extremely low-resource (ELR) settings will not work due to over-fitting. Table 1 gives the BLEU scores (Papineni et al., 2002) for Vietnamese–English (Vi–En) and Hindi–English (Hi–En) translation tasks in the Asian Languages Treebank (ALT) (Riza et al., 2016),[1] where Transformer Base models (Vaswani et al., 2017) with 1, 2, 3, and 6 encoder and decoder layers were trained on the ALT training data of 18k sentence pairs. It is clear that there is a huge performance gap between the

[1] http://www2.nict.go.jp/astrec-att/member/mutiyama/ALT/ALT-Parallel-Corpus-20191206.zip

Proceedings of the 5th Conference on Machine Translation (WMT), pages 492–502
Online, November 19–20, 2020.

#Layer	Vi→En		En→Vi		Hi→En		En→Hi	
	G	B	G	B	G	B	G	B
1	14.6	15.4	19.0	20.5	9.7	10.1	10.8	11.7
2	16.4	17.6	21.1	22.7	10.9	12.1	12.3	13.5
3	16.4	18.1	22.1	23.7	11.6	12.7	12.6	13.8
6	19.4	20.5	24.0	25.2	14.2	15.2	15.0	16.3

Table 1: The impact of number of encoder-decoder layers on translation quality: BLEU scores of greedy (G) and beam search (B, beam size of 4).

6-layer models and shallower ones. Thus, distilled corpora generated using shallower models will certainly hurt the translation quality of compact models. However, when we used the 6-layer models to generate distilled corpora, the translations were almost identical to the reference translations (with almost 100 BLEU). The reason is over-fitting despite the use of classic regularization methods such as dropout. Consequently, we need to rely on helping corpora through transfer learning (TL) methods. TL can be used by itself to improve the performance in ELR settings regardless of model efficiency. However, very little is known about how SD and TL work together.

In this paper, we investigate how to train efficient (compact and fast) NMT models for ELR settings with helping corpora through domain adaptation or cross-lingual TL. We use TL twice: once for distilling ELR corpora and then for training efficient models. We expect that training multi-domain or cross-lingual models by simply concatenating, without oversampling, the ELR corpora with the helping corpora leads to NMT models that can help generate useful distilled corpora.

To evaluate the effect of our proposed method, we experimented with two ELR language pairs, Vietnamese–English and Hindi–English (4 translation directions), in the ALT dataset. When we trained compact NMT models with 40% fewer parameters only on the distilled ELR corpus, the resulting models showed improved translation quality with greedy search by 3.6 BLEU points in average over the models trained on the original ELR corpus, while reducing 40% of decoding time. Furthermore, when we jointly used the distilled ELR corpora with the helping corpora via TL, the quality of the resulting compact models was further improved by up to 3.7 BLEU points over the best score achieved by using no distilled data. This highlights the importance of stage-wise application of SD and TL for efficient NMT models in ELR settings with high translation quality. Although the individual techniques utilized in this work are not novel, their combination and our empirical observations pertaining to the development of efficient models for ELR settings are novel.

The contributions of our paper are as follows:

- An empirical study of the combination of TL methods and SD for efficient NMT modeling.
- A cost-benefit analysis of efficient models for ELR settings.

2 Related Work

Our work is at the intersection of knowledge distillation (Hinton et al., 2015) and transfer learning for training compact NMT models.

2.1 Sequence Distillation

Knowledge distillation for sequence-to-sequence models have been successful in training efficient (compact and fast) NMT models. Sequence distillation (SD) (Kim and Rush, 2016) for NMT is a simple approach which involves training a large NMT model on a parallel corpus, translating the source side of the corpus, and then using the pseudo-parallel corpus of the same source side and the generated pseudo-target, called distilled corpus, to train a compact NMT model. The pseudo-targets represent the large model's interpretation of the original targets and can be considered as smoothed label sequences. The sequences are simpler and hence easier for smaller models to learn. As our focus is on a simple and efficient solution for ELR settings, we decided to focus only on SD.

However, its impact on ELR settings is uncertain. Given that only few thousands of domain-specific sentences are available, training large NMT models tends to over-fit on the small corpora while compact NMT models will only lead to pseudo-targets of poor quality, both preventing the generation of useful distilled corpora. It is certainly possible to search for an optimal model size. However, it will involve a time-consuming hyper-parameter search, while the result may be specific to given corpora.

2.2 Transfer Learning

Transfer learning (TL) can be in the form of domain adaptation (Chu et al., 2017) or cross-lingual or multilingual transfer (Firat et al., 2016; Zoph et al., 2016; Dabre et al., 2019; Johnson et al., 2017; Dabre et al., 2020) using helping bilingual corpora.

Assume that L_1–L_2 is an ELR language pair and L_3–L_4 is a helping pair. The given parallel corpora

for the two pairs may belong to different domains. Typically, pre-training a model on the larger L_3–L_4 corpus and then **fine-tuning** (**"ft"**) it on the smaller L_1–L_2 corpus is known to give the best translation quality for the L_1–L_2 pair (Zoph et al., 2016; Chu et al., 2017; Dabre et al., 2019), regardless of the number of model parameters. However, without careful regularization, this will definitely lead to the L_1–L_2 corpus being memorized. To address this, joint training of an NMT model using the following two methods on both corpora has been studied:

Mixed Training ("mxt"): Directly train on the concatenated corpus.

Mixed Fine-Tuning ("mxft"): First train on the L_3–L_4 corpus as in "ft," but perform fine-tuning on the concatenated corpus.

Prior to concatenating two corpora, the L_1–L_2 corpus is typically oversampled so that its size matches to the L_3–L_4 corpus. Also, we can prepend the source sentences with two artificial tokens, one indicating the domain of the corpus (Chu et al., 2017), and another indicating the target language into which we want to translate (Johnson et al., 2017). Note that when L_2 and L_4 are the same, the target language tokens are unnecessary. If L_1 and L_3 are also the same, then we are essentially performing domain adaptation.

2.3 Other Related Work

Some recent work tackled efficient NMT modeling in low-resource settings (Goyal et al., 2020; Gordon and Duh, 2020). Whereas they focus on applications of TL for compact models as this paper, there are some key differences between them and ours. Gordon and Duh (2020) focus on low-resource settings, but our low-resource data are significantly smaller than theirs. Second, whereas they use distillation twice and TL once, we recommend distillation once and TL twice. Finally, they do not examine cross-lingual TL for model compression. Goyal et al. (2020) focus on cross-lingual learning, but their approaches are centered more on leveraging orthographic or linguistic similarity, whereas we make no efforts towards orthographic unification. We thus consider parts of these studies to be orthogonal to ours.

Apart from domain adaptation and cross-lingual TL methods, low-resource settings can benefit from monolingual data, for instance, through back-translation (Sennrich et al., 2016), where target language monolingual data are translated into pseudo-source sentences. Recently, pre-training on monolingual data (Devlin et al., 2019; Song et al., 2019; Mao et al., 2020) has been proven to significantly improve the translation quality of ELR settings. Approaches involving helping monolingual data are usually more time-consuming than those that use helping bilingual corpora. Furthermore, given that our approach already needs a reasonable amount of time due to the application of TL and forward-translation of the source sentences of the parallel corpora for distilling them, we consider that such approaches should be used when no more gains can be obtained from helping bilingual corpora. We refer interested readers to work on distillation using unsupervised methods (Sun et al., 2020).

Independent of the application of TL, there exist methods for speeding up NMT, such as weight pruning (See et al., 2016) where model weights close to zero are pruned out, quantization (Lin et al., 2016) where weights are represented by faster to process integers instead of floating point numbers, aggressive model binarization (Courbariaux et al., 2017), and binary code prediction softmax (Oda et al., 2017) where the softmax is sped up by making it predict a binary code representing words instead of one-hot vectors. We expect these methods to further speed up the models obtained using our proposed method.

3 Our Approach: Transfer-Generate-Transfer

Refer to Figures 1 and 2 for a visual overview of our approaches. Figure 1 depicts the application of TL to generate distilled corpora for the ELR settings. Figure 2 depicts how the distilled ELR corpora can be used with the distilled or non-distilled helping corpora to train compact models. Our method for training compact NMT models for ELR settings can be summarized as follows:

1. Train a large joint NMT model using "**mxt**" or "**mxft**" on the concatenation of L_1–L_2 and L_3–L_4 corpora without oversampling L_1–L_2.
2. Use the joint NMT model to decode L_1 into pseudo-L_2 (L_2') and to decode L_3 into pseudo-L_4 (L_4').[2]

[2]Instead, a unidirectional $L_3 \rightarrow L_4$ model can be used to distill the L_3–L_4 corpus, because NMT models trained on the larger corpus will prevent from over-fitting and thereby generate reliable distilled data for this pair.

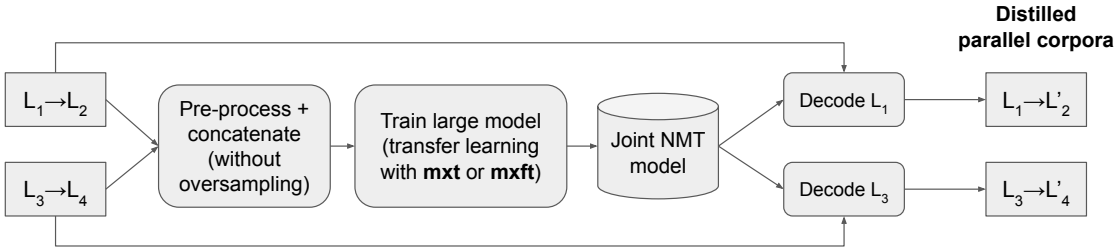


Figure 1: First round of transfer learning: training a joint model to distill the parallel corpus for extremely low-resource language pair (L_1–L_2) by leveraging a helping parallel corpus (L_3–L_4).

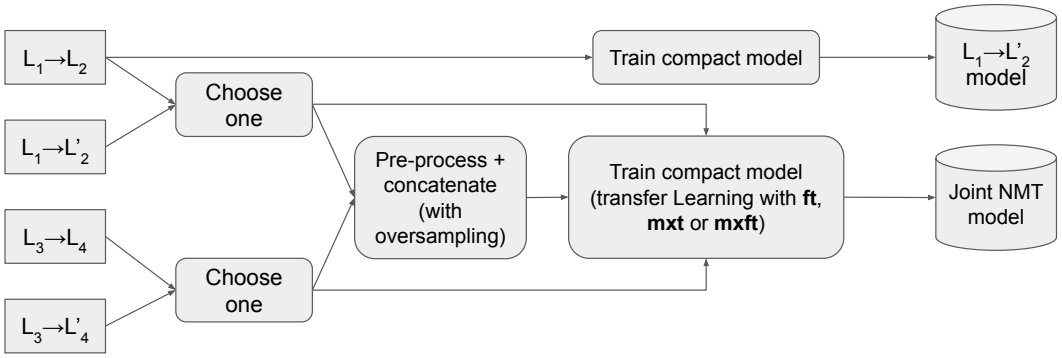


Figure 2: Second round of transfer learning: training an efficient NMT model for extremely low-resource language pairs (L_1–L_2) by leveraging a helping parallel corpus (L_3–L_4), using data distilled via the method in Figure 1. There are four possible ways of combining low-resource and helping corpora as each of them can be either distilled or non-distilled.

3. Train compact NMT models only on the L_1–L_2' corpus, or together with the distilled or non-distilled helping corpora using "**ft**," "**mxt**," or "**mxft**."

Standard TL takes place when both the ELR and helping corpora are non-distilled. In this case, TL is not used to distill data, and the ELR corpus should be oversampled to match the size of the helping corpus to ensure the best translation quality. However, for the purposes of distillation, unlike previous work, we do not oversample the L_1–L_2 corpus before concatenating it with the L_3–L_4 corpus. We did so because our preliminary explorations revealed that oversampling causes the model to memorize the L_1–L_2 corpus, thereby preventing the generation of useful distilled corpora. Naturally, the lack of oversampling might negatively impact on the quality of distilled L_1–L_2' corpus. One can empirically determine an optimal oversampling rate, but we decided to not search for it in order to make our method simple. We address this point in Section 5.1.3 with empirical evidence justifying our choice.

Note that one can pre-train compact NMT models on helping corpora and then fine-tune them on ELR corpora, avoiding SD altogether. However, the quality of TL is proportional to the quality of the pre-trained model, which tends to be high when using larger models. Furthermore, distilled data is prone to be simpler than the original data and thus has higher potential for leading to compact models. We hypothesize that distilling ELR corpora might help in better model compression. We test this hypothesis through experiments.

4 Experimental Settings

To determine the feasibility of the proposed method, we trained and evaluated NMT models in the following two groups of settings.

#1. With only distilled ELR corpora: To determine the impact of different TL settings on the quality of distilled ELR corpora and hence the compact models trained.

#2. With ELR and helping corpora: To determine the settings using both ELR and helping corpora that give compact models with highest possible translation quality.

4.1 Datasets

We experimented with the ELR Vietnamese–English (Vi–En) and Hindi–English (Hi–E) pairs from the Asian Languages Treebank (ALT) with 18,088 training, 1,000 development, and 1,018 test sentence pairs. As for the helping corpora, we used the training part of the IWSLT 2015 Vietnamese–English[3] and the IITB Hindi–English (Kunchukuttan et al., 2018),[4] consisting of 133k and 1.5M lines, respectively. We chose large as well as small helping corpora in order to determine the impact of helping corpora sizes on the model training.

4.2 Implementation Details

We used the Transformer model for our experiments (Vaswani et al., 2017) because it gives the state-of-the-art results for NMT. We made necessary changes to the code in the tensor2tensor v1.14 implementation of the Transformer in order to construct joint sub-word vocabularies as well as to handle oversampling. Tensor2tensor has its own default sub-word vocabulary learning method which we use as is by feeding it the surface word vocabulary list obtained from combining the ALT language pair and the helping language pair vocabularies. We used the default hyper-parameter setting[5] corresponding to "*transformer_base_single_gpu*" and separate source and target sub-word vocabularies of size 8,000. We chose small vocabularies as they are known to give better results for ELR settings by eliminating vocabulary sparsity. Small vocabularies also lead to models with smaller and faster softmax layers which is crucial for model compactness and speed.

We trained our models, evaluating them on the development set BLEU score every 1,000 iterations, and terminated training after 500,000 iterations or when the BLEU score did not change by more than 0.1 BLEU points for 10,000 iterations.

After training, we averaged the final 10 checkpoints to yield a single model for decoding. For decoding the test sets for evaluation, we compared greedy search and beam search with a beam size of 4, using a length penalty (alpha) of 0.6. On the other hand, for decoding the source sentences of the training sets for distillation, we only used beam search with the same beam size.

4.3 Models Evaluated

Our primary goal is to reduce the decoding time while achieving better translation quality than baselines. Following Kim and Rush (2016), who have shown that the number of encoder-decoder layers (L) have a significantly larger impact on decoding speed than hidden sizes (H), we mostly focus on compact models that use fewer encoder-decoder layers. Nevertheless, we also examine smaller hidden sizes in some experiments.

We trained simple baseline models from scratch with 1, 2, 3, and 6 layers only on the ALT training data (see Table 1).

4.3.1 Models for Distilling Corpora

To train joint models for each translation direction that is later used for distilling training data, we disjointly used the helping Vi→En, En→Vi, Hi→En, or En→Hi corpora. As we used separate source and target vocabularies and hence embedding layers, settings with a helping corpus for different translation direction can be a reasonable simulation of cross-lingual TL settings.

For joint training, we compared "**mxft**" and "**mxt**." We also considered the impact of using the domain indicator tokens (Chu et al., 2017). Thus, for each ELR and helping corpora combination, there were four types of joint models, and thus four different versions of distilled data.

4.3.2 Compact NMT Models for ELR Settings

We trained two types of models, ones that use only the distilled ELR corpora and ones that use the ELR as well as helping corpora.

#1. With only distilled ELR corpora: For each of the four helping corpora per translation direction that are used to distill data, we trained models with $L \in \{1, 2, 3\}$ and $H = 512$.[6] Additionally, we trained 3-layer models with $H \in \{128, 256\}$ to further study the tradeoff between model size and translation quality.

#2. With ELR and helping corpora: For each combination of translation direction and helping direction, we first determined the best distilled ELR corpus among four variants on

[3] https://github.com/stefan-it/nmt-en-vi

[4] http://www.cfilt.iitb.ac.in/iitb_parallel/

[5] The important hyper-parameters that remained constant throughout our experiments are: dropout of 0.1, ADAM optimizer with an initial learning rate of 0.1, 16,000 warm-up steps followed by decay for the learning rate, 8 attention heads, and a batch-size of 1,024.

[6] Feed-forward layer filter sizes were always 4 times the model's hidden size throughout this paper.

Model	Vi→En (VE)				En→Vi (EV)				Hi→En (HE)				En→Hi (EH)			
	VE	EV	HE	EH	VE	EV	HE	EH	VE	EV	HE	EH	VE	EV	HE	EH
$L=1$ $H=512$	19.7	18.9	17.0	12.9	24.9	25.4	20.9	21.1	12.9	12.1	13.8	8.5	14.5	14.2	12.2	14.6
$L=2$ $H=512$	21.9	20.8	18.6	13.6	26.7	28.0	22.2	22.7	14.9	14.2	16.3	9.3	16.7	16.2	13.4	16.9
$L=3$ $H=512$	**23.2**	22.0	18.9	14.1	28.5	**29.2**	22.7	23.1	15.7	15.0	**16.7**	9.6	17.7	16.4	13.9	**18.0**
$L=3$ $H=256$	21.1	20.1	17.7	13.5	26.6	27.3	22.1	22.4	14.1	13.5	15.7	8.7	15.4	15.6	13.5	16.2
$L=3$ $H=128$	19.4	18.2	16.7	12.5	24.5	25.4	21.3	21.1	12.2	11.4	13.7	8.4	12.8	13.6	12.2	14.8
TT	mxt	mxt	mxft	mxt	mxft	mxft	mxft	mxft	mxt	mxt	mxt	mxft	mxft	mxt	mxft	mxt
DT	yes	yes	yes	no	yes	yes	yes	no	yes	no	yes	no	yes	no	yes	yes
$L=6$ $H=512$ (see Table 1)	greedy: 19.4 beam: 20.5				greedy: 24.0 beam: 25.2				greedy: 14.2 beam: 15.2				greedy: 15.0 beam: 16.3			

Table 2: BLEU scores for each ELR translation task achieved by our proposed method with greedy search. The second row indicates the translation direction of helping data. The highest scores for each translation direction are highlighted in bold. "TT" and "DT" respectively represent the type of training ("mxt" or "mxft") and whether domain tags were used ("yes" or "no") for the joint training that led to the best distilled corpora. The last row shows the greedy and beam search BLEU scores of the baseline 6-layer models for comparison (see Table 1).

the basis of BLEU score of the $L = 3$ and $H = 512$ model trained only on it (#1), and then combined it with the distilled helping corpus to train models with $L \in \{1, 2, 3\}$ and $H = 512$ using "**ft**," "**mxt**," and "**mxft**." We also trained models with the same configurations for combinations of ELR and helping corpora where only one of the corpora are distilled. As (strong) baselines, we trained models with $L \in \{1, 2, 3, 6\}$ and $H = 512$ trained on non-distilled ELR and helping corpora.

5 Results

We show results using only distilled ELR corpora and then using it with helping corpora.

5.1 Using Only Distilled ELR Corpora

In Table 2, we show how domain adaptation and cross-lingual TL methods affect creation of distilled ELR corpora and hence the greedy search translation quality of efficient models. Greedy search is emphasized due to our focus on fast decoding speed as well as high translation quality.

5.1.1 Translation Quality of Efficient Models

For each translation direction, the best distilled corpora used to train models with 3 layers gives greedy search translation quality ranging from 1.5 to 4.0 BLEU points over the 6-layer non-distilled baseline model's beam search translation quality. Comparing the 1-, 2-, and 3-layer models trained with the best distilled corpora with their non-distilled counterparts in Table 1, we can see that there is an improvement of 2.9 to 5.5 BLEU points. Considering that we used the distilled equivalents of the original training data, this result shows the explicit effect of TL and SD which helps generate data that improves translation quality despite reducing the model size.

Training models on ELR corpora can finish quickly. Thus, our distilled corpora can be used in situations where quick deployment of compact and fast NMT models is important.

5.1.2 Domain Adaptation vs. Cross-Lingual Transfer

Our experiment revealed that cross-lingual training is definitely a viable alternative. For instance, in Vi→En translation, the best BLEU score was achieved when the helping direction was also Vi→En. When the helping direction was Hi→En, these improvements were much smaller. Nevertheless, it is clear that cross-lingual training is useful when domain adaptation is not possible. Work on script mapping to improve the quality of TL (Song et al., 2020; Goyal et al., 2020) indicates that our cross-lingual distillation procedure might give better results if we mapped Hi to Vi or vice-versa. We leave this for future work.

Consider two hypothetical settings for Vi→En translation, where we used the reversed, En→Vi and En→Hi, helping directions to generate distilled corpora for Vi→En translation. When using En→Vi as the helping direction, the BLEU scores of greedy search with 1-, 2-, and 3-layer models improved by 4.3, 4.4, and 5.6 BLEU points, respectively. These improvements are approximately 1.0 BLEU points lower than those obtained in the domain adaptation setting with Vi→En as the helping direction, but it shows that using helping corpora with different languages can be of some use. However, when using En→Hi as the helping direction, the BLEU scores dropped. Note that English

DT	TT	Vi→En	En→Vi	Hi→En	En→Hi
yes	mxft	21.5	**29.2**	15.3	15.4
yes	mxt	**23.2**	28.0	**16.7**	**18.0**
no	mxft	21.9	29.1	14.2	12.9
no	mxt	21.1	28.0	16.2	16.1

Table 3: Impact of domain tags (DT) and training type (TT) on the greedy search translation quality (BLEU) of models with $L = 3$ and $H = 512$. The best scores are in bold.

Size	Vi→En		En→Vi		Hi→En		En→Hi	
	HE	EH	HE	EH	HE	EH	HE	EH
133k	20.5	19.2	26.4	25.9	15.2	14.6	16.4	17.1
200k	21.3	**20.2**	27.1	28.0	16.3	15.2	16.4	17.1
500k	**21.5**	19.9	27.1	**28.2**	**17.2**	**15.8**	**17.5**	17.5
1500k	18.9	14.1	**22.7**	23.1	16.7	9.6	13.9	**18.0**

Table 4: Impact of helping corpus size on the greedy search translation quality (BLEU) for each translation task achieved with models with $L = 3$ and $H = 512$. The best scores are in bold.

and Vietnamese use the Roman alphabet which might enable cognate sharing even when the ELR and helping directions are opposite. However, this is not fully applicable when En→Hi is the helping direction. Furthermore, the Hindi–English corpus was much larger than the one for Vietnamese–English. Since we do not oversample the ELR corpora for distilling corpora, we expect that the model heavily focuses on the Hindi–English pair which could negatively impact on the quality of the resulting distilled corpora.

While similar observations are applicable to other translation directions, consider Hi→En and En→Hi translation. As before, using Hi→En and En→Hi helping directions respectively using domain adaptation resulted in the best distilled corpora. However, using the reverse En→Hi and Hi→En helping directions, respectively, led to a drop in translation quality. In contrast, using Vi→En and En→Vi helping directions led to distilled corpora that led to compact models giving translations within 1.0 BLEU points of those given by the best distilled corpora. This shows that in a cross-lingual TL setting for distilling ELR corpora, it may be better to have helping corpora that are not much larger than the ELR corpora. We validate this hypothesis in Section 5.1.3.

As for the use of domain indicator tags, 11 out 16 cases indicate that such tags are useful. In Table 3, we show the results of model with $L = 3$ and $H = 512$ trained on distilled data generated with and without domain indicator tags when training using "mxt" and "mxft" (4 combinations). For simplicity, we show results for when the ELR and helping directions are the same. Using domain tags gives better results when the helping corpora are substantially larger than the ELR corpora. But when the helping corpora are relatively smaller (Vietnamese–English), domain tags do not seem to have a large impact. Furthermore, "mxt" tends to be better than "mxft." Overall, simply concatenating the ELR and helping corpora without oversampling or domain indicators and then training joint model in one stage should be sufficient to yield useful distilled corpora. We will experiment with additional language pairs and domains in the future to conclusively determine a one-fits-all setting.

5.1.3 Impact of Helping Corpora Size

We observed that a large helping corpus degrades the translation quality in cross-lingual settings. Instead of determining an optimal oversampling ratio for the ELR corpus, we experimented with downsampling the helping corpus size. We did this to avoid running into the risk of over-fitting due to oversampling. We experimented with the downsampled versions of the Hindi–English corpus: we prepared sub-corpora with 500k, 200k, and 133k sentence pairs, assuring that a larger one subsumes all the smaller ones. For simplicity, we reused the best configurations reported in Table 2.

Table 4 shows the greedy search results. When using the entire Hindi–English helping corpus for Vi→En and En→Vi translation tasks, the BLEU score is substantially lower than the baseline models, indicating the poor quality of the distilled data. Note that we do not oversample the ELR corpora for distillation and thus coupling them with a larger helping corpus is detrimental to the final translation quality, as the NMT model sees more examples in the latter than the former. However, using significantly smaller corpora ensures that the NMT model sees much fewer examples in the helping corpus and thus is able to better learn from the ELR corpus leading to better distilled data. This is evidenced by the improved BLEU scores when using downsampled helping corpora. Naturally, using the Vi→En helping corpus gives the best results for Vi→En translation tasks, but the results using the downsampled Hindi–English helping corpora are within 2.0 BLEU points of the best. Note also that the BLEU score for Hi→En task using a helping corpus with 500k sentence pairs (17.2) surpasses the

Model		Size	Time	BLEU
$L=1$	$H=512$	19.0M	11.7s	18.4
$L=2$	$H=512$	27.0M	17.6s	20.8
$L=3$	$H=512$	34.0M	22.5s	21.8
$L=3$	$H=256$	11.0M	22.0s	18.8
$L=3$	$H=128$	4.0M	21.3s	17.3
$L=6$ (see Table 1)	$H=512$	56.6M	37.6s	18.2

Table 5: Comparison of size, decoding time (with greedy search), and BLEU score for various models evaluated in Table 2. For each column, average value for four translation directions is reported.

score obtained using all the sentence pairs (16.7) by 0.5 BLEU points. For the reverse direction, the score (17.5) is within 0.5 BLEU points of the best score (18.0). It is clear that choosing an appropriate helping corpus size is important for generating useful distilled corpora. This result further reinforces our claim that cross-lingual training is a viable option for generating useful distilled data. Such cross-lingual training also has the potential to distill data that can help train compact models with BLEU score higher than larger models trained on non-distilled data. As for optimal size of helping corpus, the performance gap between using 200k and 500k helping sentence pairs is very small in most settings. This means that distilling data does not need too much helping corpus and thus in practice choosing a small sample of the helping corpus can help significantly save time for model training and subsequent corpus distillation. This also helps avoid the issue of oversampling and thereby maintaining the simplicity of the method.

A fair comparison with the same size (133k) of helping corpora confirmed that sharing at least one of source and target languages tends to improve the final translation quality in cross-lingual TL settings. For instance, Hi→En has a better impact than En→Hi on Vi→En. Similarly, En→Hi leads to higher BLEU score than Hi→En for En→Vi.

5.1.4 Size vs. Speed vs. Translation Quality

Table 5 compares size, decoding time, and BLEU score for various models. As the model size drops with fewer layers and smaller hidden sizes, BLEU score also drops. However, the decoding time decreases significantly. Note that reducing the number of layers mainly impacts on the decoding time, whereas reducing hidden sizes does not have such a huge impact, as reported in Kim and Rush (2016).

We observed that the model with $L = 3$ and $H = 512$ are approximately 1.7 times (or 40%) smaller and 1.7 times (40%) faster than the 6-layer models despite exhibiting improved translation quality of 3.6 BLEU points in average. If one wishes to save decoding time, we suggest to train a model with $L = 1$ and $H = 512$, which is approximately 3.0 times smaller and 3.2 times faster than a 6-layer model, while having comparable translation quality. If the priority is reducing model size, then using models with $L = 3$ and $H \in \{256, 128\}$ are 5.1 times to 14.2 times smaller, even though they do not benefit much from narrowing down H. The model with $L = 3$ and $H = 256$ is comparable to the one with $L = 1$ and $H = 512$ in terms of quality, but the latter is 1.7 times smaller than the former. We recommend experimenting with different model sizes before choosing the best one for the target application.

5.2 Using Both ELR and Helping Corpora

Table 6 gives the BLEU scores achieved by models trained on both ELR and helping corpora, where we compare the distilled ("Y") and non-distilled ("N") versions of corpora as well as the three types of training ("ft," "mxt," and "mxft").

5.2.1 Importance of Transfer Learning for Efficient Models

Comparing the results of using only ELR corpora against the results of TL without SD, TL already gives 1-layer models that are competitive, if not better than the 3-layer models trained on non-distilled ELR corpora and the 6-layer models trained on distilled ELR corpora. The 1-layer models are 1.9 and 3.2 times faster as well as approximately 1.8 and 3.0 times smaller than the 3- and 6-layer models, respectively (see Table 5). It is thus reasonable to avoid SD altogether when time is of the essence.

Among the training methods, "mxft" was in most cases slightly better than "ft" and both of them are substantially better than "mxt." This highlights the importance of stage-wise TL rather than innocently training on a combination of all corpora. Note that "mxt" achieved the highest BLEU score for some configurations, and it should be a reasonable option when there is not enough time for stage-wise training.

5.2.2 Importance of Distillation with Transfer Learning for NMT Efficiency

Using at least one distilled corpus, either ELR or helping corpora, is important in improving the translation quality of compact models. For instance,

ELR	HD	TT	Vi→En								Hi→En							
			$L=1$		$L=2$		$L=3$		$L=6$		$L=1$		$L=2$		$L=3$		$L=6$	
			G	B	G	B	G	B	G	B	G	B	G	B	G	B	G	B
N	-	-	14.6	15.4	16.4	17.6	16.4	18.1	19.4	20.5	9.7	10.1	10.9	12.1	11.6	12.7	14.2	15.2
Y	-	-	19.7	19.9	21.9	22.5	23.2	23.1	-	-	13.8	14.4	16.3	16.9	16.7	17.4	-	-
N	N	ft	21.5	22.8	24.6	25.9	25.6	26.8	26.6	27.5	19.5	20.7	25.0	26.1	26.4	27.2	28.1	29.0
		mxt	20.3	22.1	23.3	25.0	24.5	26.0	26.1	27.5	15.0	15.9	18.1	19.9	20.9	22.3	23.3	23.7
		mxft	21.4	22.9	26.2	27.3	26.7	28.0	**27.7**	**28.6**	19.9	20.9	25.4	26.3	27.8	28.7	**29.3**	**29.8**
N	Y	ft	21.9	23.4	25.0	26.3	26.1	27.1	-	-	20.9	21.9	25.9	**27.5**	27.8	28.8	-	-
		mxt	20.9	22.8	24.0	25.4	24.4	25.9	-	-	16.9	18.1	20.6	21.7	22.4	23.1	-	-
		mxft	22.3	24.3	26.4	27.7	26.4	27.6	-	-	21.5	**22.2**	**26.8**	**27.5**	**28.1**	**29.0**	-	-
Y	N	ft	24.1	24.3	26.7	26.9	27.3	27.7	-	-	20.5	20.2	24.7	24.6	25.9	25.9	-	-
		mxt	24.3	25.1	26.8	27.1	27.5	28.1	-	-	18.2	19.2	22.9	23.7	24.0	24.6	-	-
		mxft	24.3	25.1	27.5	**28.3**	27.6	28.0	-	-	20.3	20.9	25.0	24.5	25.9	24.5	-	-
Y	Y	ft	25.1	25.7	27.0	27.7	27.9	28.4	-	-	21.5	21.7	26.0	26.3	26.5	26.7	-	-
		mxt	24.8	25.5	**27.8**	28.1	**28.2**	28.8	-	-	19.8	20.1	24.2	24.7	25.4	25.6	-	-
		mxft	**25.2**	**25.8**	27.6	28.2	28.1	**28.9**	-	-	**21.6**	21.9	26.4	25.8	26.9	25.8	-	-

Table 6: BLEU scores for Vi→En and Hi→En translation tasks with greedy (G) and beam search (B). Models trained on either distilled ("Y") or non-distilled ("N") version of ELR and helping corpora ("ELR" and "HD" columns, respectively) using different domain adaptation techniques ("TT" column), are compared. The highest score(s) in each column are marked in bold.

the BLEU score of greedy search with the 1-layer models trained on some distilled data are up to 3.7 BLEU points higher than the best scores achieved by 1-layer models that do not use distilled data at all (21.5 and 19.9 for Vi→En and Hi→En by "N–N" models in Table 6, respectively). Although the gap between the performances tends to be narrower when the number of layers increases, this sacrifices compactness and decoding speed.

The behavior of models trained on distilled data differs depending on the combination of ELR and helping corpora. For Vi→En, distilling the ELR corpus ("Y–N") is more useful than distilling the helping corpus ("N–Y"). In contrast, for Hi→En, distilling the helping corpus ("N–Y") matters more. Recall that the Vi→En helping corpus is around 10 times smaller than the Hi→En helping corpus. This means that a compact model has to bear the burden of learning a much larger amount of knowledge from the larger helping corpus. Consequently, the compact model should be better at learning the Vi→En helping corpus, especially in its distilled form. Furthermore, given that the distilled ELR corpus for Vi→En already improves translation quality compared to its non-distilled counterpart, it should also help improve translation quality when used it in combination with the helping corpus. This is indicated by the best result for Vi→En achieved by distilling both the ELR and helping corpora. For this direction, the 2-layer models trained on distilled data are either competitive with if not better than the 6-layer models. For Hi→En, given that the size of helping corpus is significantly larger, distilling it into compact models is harder due to lack of parameters. This is the most likely reason behind the relatively small improvement by distilled data. Although the impact of SD on TL on Hi→En is not as impressive as for Vi→En, we advise experimenting with SD rather than not.

6 Conclusion

In this paper, we have explored the combination of transfer learning (TL) and sequence distillation for obtaining compact and fast models in extremely low-resource (ELR) settings. Our experiments on four translation directions revealed that leveraging helping corpora help in distilling ELR corpora that help train compact models with 3.6 average BLEU points improvement in translation quality. Compact models trained on distilled ELR corpora are not only fast but also give better translations than larger models trained on non-distilled ELR corpora. We showed the effects of choosing appropriate training methods, using domain indicator tags, and managing corpora sizes on translation quality. Our cost-benefit analysis of model size, decoding speed, and translation quality showed that we can achieve translation quality comparable to baselines trained on the original ELR corpora with models that are approximately 3.0 times smaller and 3.2 times faster than said baselines. We also showed that combining distilled ELR corpora with the distilled or non-distilled helping corpora, using simple TL methods, can further boost the performance of compact and hence fast NMT models. We strongly recommend to leverage distilled ELR

corpora through stage-wise TL for compact and high-quality NMT for ELR settings.

In our future work, we will extend our approach for a single compact multilingual NMT model, for instance, focusing on multi-parallel ALT dataset.

Acknowledgments

A part of this work was conducted under the commissioned research program "Research and Development of Advanced Multilingual Translation Technology" in the "R&D Project for Information and Communications Technology (JPMI00316)" of the Ministry of Internal Affairs and Communications (MIC), Japan. Atsushi Fujita was partly supported by JSPS KAKENHI Grant Number 19H05660.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, San Diego, USA. International Conference on Learning Representations.

Chenhui Chu, Raj Dabre, and Sadao Kurohashi. 2017. An empirical comparison of domain adaptation methods for neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 385–391, Vancouver, Canada. Association for Computational Linguistics.

Matthieu Courbariaux, Itay Hubara, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. 2017. Binarized neural networks: Training deep neural networks with weights and activations constrained to +1 or -1. *CoRR*, abs/1602.02830.

Raj Dabre, Chenhui Chu, and Anoop Kunchukuttan. 2020. A comprehensive survey of multilingual neural machine translation.

Raj Dabre, Atsushi Fujita, and Chenhui Chu. 2019. Exploiting multilingualism through multistage fine-tuning for low-resource neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1410–1416, Hong Kong, China. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, USA. Association for Computational Linguistics.

Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. 2016. Multi-way, multilingual neural machine translation with a shared attention mechanism. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 866–875, San Diego, USA. Association for Computational Linguistics.

Mitchell Gordon and Kevin Duh. 2020. Distill, adapt, distill: Training small, in-domain models for neural machine translation. In *Proceedings of the Fourth Workshop on Neural Generation and Translation*, pages 110–118, Online. Association for Computational Linguistics.

Vikrant Goyal, Sourav Kumar, and Dipti Misra Sharma. 2020. Efficient neural machine translation for low-resource languages via exploiting related languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 162–168, Online. Association for Computational Linguistics.

Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531.

Melvin Johnson, Mike Schuster, Quoc Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernand a Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327, Austin, USA. Association for Computational Linguistics.

Anoop Kunchukuttan, Pratik Mehta, and Pushpak Bhattacharyya. 2018. The IIT Bombay English-Hindi parallel corpus. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Darryl D. Lin, Sachin S. Talathi, and V. Sreekanth Annapureddy. 2016. Fixed point quantization of deep convolutional networks. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning*, pages 2849–2858, New York, USA.

Zhuoyuan Mao, Fabien Cromieres, Raj Dabre, Haiyue Song, and Sadao Kurohashi. 2020. JASS: Japanese-specific sequence to sequence pre-training for neural machine translation. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages

3683–3691, Marseille, France. European Language Resources Association.

Yusuke Oda, Philip Arthur, Graham Neubig, Koichiro Yoshino, and Satoshi Nakamura. 2017. Neural machine translation via binary code prediction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 850–860, Vancouver, Canada. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL)*, pages 311–318, Philadelphia, USA. Association for Computational Linguistics.

Hammam Riza, Michael Purwoadi, Gunarso, Teduh Uliniansyah, Aw Ai Ti, Sharifah Mahani Aljunied, Luong Chi Mai, Vu Tat Thang, Nguyen Phuong Thai, Vichet Chea, Rapid Sun, Sethserey Sam, Sopheap Seng, Khin Mar Soe, Khin Thandar Nwet, Masao Utiyama, and Chenche n Ding. 2016. Introduction of the Asian Language Treebank. In *Proceedings of the 2016 Conference of the Oriental Chapter of International Committee for Coordination and Standardization of Speech Database s and Assessment Technique (O-COCOSDA)*, pages 1–6, Bali, Indonesia.

Abigail See, Minh-Thang Luong, and Christopher D. Manning. 2016. Compression of neural machine translation models via pruning. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, pages 291–301, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL), Volume 1: Long Papers*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Haiyue Song, Raj Dabre, Zhuoyuan Mao, Fei Cheng, Sadao Kurohashi, and Eiichiro Sumita. 2020. Pre-training via leveraging assisting languages for neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 279–285, Online. Association for Computational Linguistics.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. MASS: masked sequence to sequence pre-training for language generation. In *Proceedings of the 36th International Conference on Machine Learning*, pages 5926–5936, Long Beach, USA.

Haipeng Sun, Rui Wang, Kehai Chen, Masao Utiyama, Eiichiro Sumita, and Tiejun Zhao. 2020. Knowledge distillation for multilingual unsupervised neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3525–3535, Online. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of the 27th Neural Information Processing Systems Conference (NIPS)*, pages 3104–3112, Montréal, Canada. Curran Associates, Inc.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 30th Neural Information Processing Systems Conference (NIPS)*, pages 5998–6008, Long Beach, USA. Curran Associates, Inc.

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer learning for low-resource neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1568–1575, Austin, USA. Association for Computational Linguistics.

Fast Interleaved Bidirectional Sequence Generation

Biao Zhang[1] **Ivan Titov**[1,2] **Rico Sennrich**[3,1]

[1]School of Informatics, University of Edinburgh
[2]ILLC, University of Amsterdam
[3]Department of Computational Linguistics, University of Zurich
B.Zhang@ed.ac.uk, ititov@inf.ed.ac.uk, sennrich@cl.uzh.ch

Abstract

Independence assumptions during sequence generation can speed up inference, but parallel generation of highly inter-dependent tokens comes at a cost in quality. Instead of assuming independence between neighbouring tokens (semi-autoregressive decoding, SA), we take inspiration from bidirectional sequence generation and introduce a decoder that generates target words from the left-to-right and right-to-left directions simultaneously. We show that we can easily convert a standard architecture for unidirectional decoding into a bidirectional decoder by simply interleaving the two directions and adapting the word positions and self-attention masks. Our interleaved bidirectional decoder (IBDecoder) retains the model simplicity and training efficiency of the standard Transformer, and on five machine translation tasks and two document summarization tasks, achieves a decoding speedup of ∼2× compared to autoregressive decoding with comparable quality. Notably, it outperforms left-to-right SA because the independence assumptions in IBDecoder are more felicitous. To achieve even higher speedups, we explore hybrid models where we either simultaneously predict multiple neighbouring tokens per direction, or perform multi-directional decoding by partitioning the target sequence. These methods achieve speedups to 4×–11× across different tasks at the cost of <1 BLEU or <0.5 ROUGE (on average).[1]

1 Introduction

Neural sequence generation aided by encoder-decoder models (Bahdanau et al., 2015; Vaswani et al., 2017) has achieved great success in recent years (Bojar et al., 2018; Song et al., 2019; Raffel et al., 2019; Karita et al., 2019), but still suffers from slow inference. One crucial bottleneck lies in its generative paradigm which factorizes the conditional probability along the target sequence $\mathbf{y} = \{y_1, y_2, \ldots, y_n\}$ of length n as follows:

$$p(\mathbf{y}|\mathbf{x}) = \prod_{t=1}^{n} p\left(y_t|\mathbf{y}_{<t}, \mathbf{x}\right), \quad (1)$$

where $\mathbf{x} = \{x_1, x_2, \ldots, x_m\}$ is the source sequence of length m. This factorization determines that target words can only be generated one-by-one in a sequential and unidirectional manner, which limits the decoding efficiency.

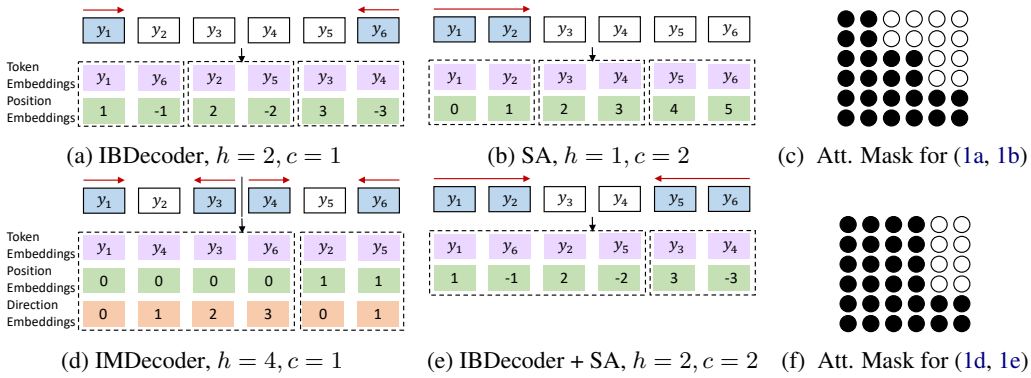
A promising direction to break this barrier is to generate multiple target words at one decoding step to improve the parallelization of inference (Gu et al., 2018; Stern et al., 2018). However, this introduces independence assumptions that hurt translation quality, since words produced in parallel are in fact likely to be inter-dependent. We hypothesize that there are groups of words that are less likely to be strongly inter-dependent than neighbouring words, which will allow for better parallelization. Inspired by bidirectional modeling (Zhang et al., 2019b, 2020), we resort to an alternative probabilistic factorization:

$$p^{\text{BD}}(\mathbf{y}|\mathbf{x}) = \prod_{t=1}^{\lceil n/2 \rceil} p\left(\overrightarrow{y_t}, \overleftarrow{y_{t'}}|\overrightarrow{\mathbf{y}_{<t}}, \overleftarrow{\mathbf{y}_{>t'}}, \mathbf{x}\right), \quad (2)$$

Introducing an independence assumption between t and $t' = n - t + 1$ allows for parallel word prediction from both the $\overrightarrow{\text{left-to-right}}$ and $\overleftarrow{\text{right-to-left}}$ directions. Based on this factorization, Zhou et al. (2019) propose synchronous bidirectional translation using a dedicated interactive decoder, and report quality improvements compared to left-to-right semi-autoregressive decoding (Wang et al., 2018, SA) in translation quality. However, their success comes along with extra computational overhead brought by the specialized decoder. Empirically, Zhou et al. (2019) only report a decoding

[1]Source code is released at https://github.com/bzhangGo/zero.

Proceedings of the 5th Conference on Machine Translation (WMT), pages 503–515
Online, November 19–20, 2020.



(a) IBDecoder, $h = 2, c = 1$ (b) SA, $h = 1, c = 2$ (c) Att. Mask for (1a, 1b)

(d) IMDecoder, $h = 4, c = 1$ (e) IBDecoder + SA, $h = 2, c = 2$ (f) Att. Mask for (1d, 1e)

Figure 1: Overview of the interleaved bidirectional decoder (IBDecoder, 1a), the semi-autoregressive decoder (SA, 1b), the interleaved multi-directional decoder (IMDecoder, 1d) and the bidirectional semi-autoregressive decoder (IBDecoder+ SA, 1e) on target sequence $\mathbf{y} = \{y_1, y_2, \ldots, y_6\}$. We reorganize the target sequence (purple), the word positions (green) and the self-attention mask (circles) to reuse the standard Transformer decoder. During inference, multiple target words are generated simultaneously at each step (dashed rectangles), improving the decoding speed. The self-attention masks are given in (1c) and (1f), where sold black circles indicate allowed attention positions. Red arrows indicate generation directions (h is the direction number), whose length denotes the number of words produced per direction (c). Blue rectangles denote words generated at the first step. The direction embedding (red rectangles) reflects the direction each target word belongs to. Apart from the left-to-right generation, IBDecoder jointly models the right-to-left counterpart within a single sequence. IMDecoder extends IBDecoder by splitting the sequence into several equal segments and performing bidirectional generation on each of them, while IBDecoder+SA allows each direction to produce multiple words.

speedup of 1.38×, slower than SA, although the factorization halves the decoding steps.

We combine the strengths of bidirectional modeling and SA, and propose interleaved bidirectional decoder (IBDecoder) for fast generation. As shown in Figure 1a, we interleave target words from the left-to-right and right-to-left directions and separate their positions to support reusing any standard unidirectional decoders, such as the Transformer decoder (Vaswani et al., 2017). We reorganize the self-attention mask to enable inter- and intra-direction interaction (Figure 1c) following SA. Unlike SA, we show through experiments that distant tokens from different directions are less inter-dependent, providing a guarantee for better performance. Compared to previous studies (Zhang et al., 2018d, 2019b, 2020; Zhou et al., 2019), our approach has no extra model parameters and brings in little overhead at training and decoding.

IBDecoder is speedup-bounded at 2×. To push this ceiling up, we explore strategies for multi-word simultaneous generation, including *multi-directional decoding* (IMDecoder, Figure 1d) and *SA* (Figure 1b). The former extends Eq. 2 by inserting more generation directions, while the latter allows each direction to produce multiple target words (Wang et al., 2018). These strategies offer us a chance to aggressively improve the decoding speed albeit at the risk of degenerated performance. To encourage multi-word generation in parallel, we propose a modified beam search algorithm.

We extensively experiment on five machine translation tasks and two document summarization tasks, with an in-depth analysis studying the impact of batch size, beam size and sequence length on the decoding speed. We close our analysis by examining the capacity of our model in handling long-range dependencies. On these tasks, IBDecoder yields ∼2× speedup against Transformer at inference, and reaches 4×–11× after pairing it with SA. Still, the overall generation quality is comparable. When we pair our method with sequence-level knowledge distillation (Kim and Rush, 2016), we outperform a Transformer baseline on 6 out of 7 tasks.

Our contributions are summarized below:

- We propose IBDecoder, following a bidirectional factorization of the conditional probability, for fast sequence generation. IBDecoder retains the training efficiency and is easy to implement.
- We extend IBDecoder to enable multi-word simultaneous generation by investigating integration with IMDecoder and SA. Results show that IBDecoder + SA performs better than IMDecoder.
- We propose a modified beam search algorithm to support step-wise parallel generation.
- On several sequence generation benchmarks, IBDecoder yields ∼2× speedup against Transformer at inference, and reaches 4×–11× af-

ter pairing it with SA. Still, the overall generation quality is comparable.

2 Related Work

Efforts on fast sequence generation come along with the rapid development of encoder-decoder models (Vaswani et al., 2017). A straightforward way is to reduce the amount of computation. Methods in this category range from teacher-student model (Kim and Rush, 2016; Hayashi et al., 2019), constrained softmax prediction (Hu et al., 2015), beam search cube pruning (Zhang et al., 2018c), float-point quantization (Wu et al., 2016; Bhandare et al., 2019), model pruning (See et al., 2016), to simplified decoder architectures, such as lightweight recurrent models (Zhang et al., 2018b; Zhang and Sennrich, 2019; Kim et al., 2019), average attention network (Zhang et al., 2018a), merged attention network (Zhang et al., 2019a), dynamic convolution (Wu et al., 2019), and hybrid attentions (Shazeer, 2019; Wang et al., 2019), .etc.

Nonetheless, the above methods still suffer from the inference bottleneck caused by the sequential nature of autoregressive models. Instead, Gu et al. (2018) propose non-autoregressive generation where target words are predicted independently, leading to great speedup, albeit at a high cost to generation quality. Follow-up studies often seek solutions to recover the performance (Libovický and Helcl, 2018; Guo et al., 2019; Shao et al., 2020; Ghazvininejad et al., 2020; Ran et al., 2020), but also reveal the trade-off between the quality and speed in terms of autoregressiveness. This motivates researchers to discover the optimal balance by resorting to semi-autoregressive modeling (Wang et al., 2018; Stern et al., 2018), iterative refinement (Lee et al., 2018; Stern et al., 2019; Ghazvininejad et al., 2019) or in-between (Kaiser et al., 2018; Akoury et al., 2019).

We hypothesize that generation order affects the felicity of independence assumptions made in semi-autoregressive modelling. Unlike generation with flexible orders (Emelianenko et al., 2019; Stern et al., 2019; Gu et al., 2019a), we employ deterministic generation order for model simplicity and training efficiency, specifically focusing on bidirectional decoding. The study of bidirectional modeling dates back to the era of phase-based statistical machine translation (Watanabe and Sumita, 2002; Finch and Sumita, 2009) and recently gained popularity in neural machine translation (Liu et al., 2016; Sennrich et al., 2016a; Zhang et al., 2019c,b; Zheng et al., 2019). Unfortunately, these methods either design complex neural decoders, which hurts training efficiency, and/or perform the left-to-right and right-to-left inference separately followed by rescoring, which slows down decoding. By contrast, our model speeds up inference while maintaining training speed.

Our work is closely related to SA (Wang et al., 2018) and synchronous bidirectional generation (Zhou et al., 2019). IBDecoder extends SA to incorporate information from different directions. In contrast to Zhou et al. (2019), we only make minimal changes to the standard Transformer decoder, which benefits efficiency during training and inference, and makes our method easy to implement. We also find improvements in both decoding speed and translation quality compared to (Wang et al., 2018; Zhou et al., 2019).

3 Autoregressive Transformer

Transformer (Vaswani et al., 2017), the state-of-the-art neural sequence generation model, follows the autoregressive factorization as in Eq. 1. To handle the dependency of target word y_t on previous target words $\mathbf{y}_{<t}$, Transformer relies on a masked self-attention network in the decoder:

$$\textsc{Att}(\mathbf{Y}^l, \mathbf{M}) = f\left(\frac{\mathbf{Q}^l {\mathbf{K}^l}^T}{\sqrt{d}} + \mathbf{M}\right)\mathbf{V}^l \quad (3)$$

where $\mathbf{Q}^l, \mathbf{K}^l, \mathbf{V}^l = \mathbf{W}_q^l\mathbf{Y}^l, \mathbf{W}_k^l\mathbf{Y}^l, \mathbf{W}_v^l\mathbf{Y}^l \in \mathbb{R}^{n\times d}$, $f(\cdot)$ denotes softmax operation, d is model dimension and l is layer depth. $\mathbf{W}_q, \mathbf{W}_k, \mathbf{W}_v \in \mathbb{R}^{d\times d}$ are trainable parameters.

The mask matrix $\mathbf{M} \in \mathbb{R}^{n\times n}$ limits the access of attention to only the past target words. Formally, given the target sequence length n, this matrix can be constructed by the following masking function:

$$\mathcal{M}_{i,j}(h,c) = \begin{cases} 0, & \text{if } \lceil i/(h\cdot c)\rceil \geq \lceil j/(h\cdot c)\rceil \\ -\infty, & \text{otherwise} \end{cases} . \quad (4)$$

where $0 < i, j < n$, h denotes the number of generation directions, and c is the number of target words predicted per direction. By default, the Transformer decoder is unidirectional and generates words one-by-one. Thus, $\mathbf{M} = \mathcal{M}(1,1)$. The infinity here forces softmax output a probability of 0, disabling invalid attentions.

The input layer to Transformer's decoder is the addition of target word embedding $\mathbf{E}_\mathbf{y}$ and word

position encoding $\text{PE}_{\mathcal{T}}$, i.e $\mathbf{Y}^0 = \mathbf{E}_{\mathbf{y}} + \text{PE}_{\mathcal{T}} \in \mathbb{R}^{n \times d}$. $\mathcal{T}$ maps $\mathbf{y}$ to its word position sequence, which is a simple indexing function (Figure 1b):

$$\mathcal{T}_t = t - 1, \tag{5}$$

where $t = 1 \dots n$. Transformer adopts the sinusoidal positional encoding to project these indexes to real-space embeddings, and uses the last-layer decoder output $\mathbf{Y}^L$ to predict the respective next target word. We explain how to accelerate generation by reordering $\mathbf{y}$, adjusting h, c and $\mathcal{T}$ next.

4 Interleaved Bidirectional Decoder

The structure of Transformer is highly parallelizable, but the autoregressive schema $(h = 1, c = 1)$ blocks this parallelization during inference. We alleviate this barrier by exploring the alternative probabilistic factorization in Eq. 2 to allow words predicted from different directions simultaneously.

We propose IBDecoder as shown in Figure 1a. We reuse the standard decoder's architecture in a bid to largely inherit Transformer's parallelization and avoid extra computation or parameters, rather than devising dedicated decoder architectures (Zhou et al., 2019; Zhang et al., 2020). To make the left-to-right and right-to-left generation collaborative, we reorganize the target sequence and the word positions below (purple and green rectangles in Figure 1a):

$$\mathbf{y}^{\text{BD}} = \left[y_1 y_n, y_2 y_{n-1}, ..., y_{\lfloor n/2 \rfloor + 1}\right], \tag{6}$$

$$\mathcal{T}_t^{\text{BD}} = (-1)^{(t-1)} \lceil t/2 \rceil. \tag{7}$$

By following the generation order defined by Eq. 2, the sequence $\mathbf{y}^{\text{BD}}$ interleaves $\mathbf{y}_{1:\lfloor n/2 \rfloor}$ and $\mathbf{y}_{\lfloor n/2 \rfloor + 1:n}$ and converts a bidirectional generation problem to a unidirectional one. We introduce negative positions to $\mathcal{T}^{\text{BD}}$ to retain the locality bias of sinusoidal positional encodings in $\mathbf{y}^{\text{BD}}$.[2] Compared to $(\mathbf{y}, \mathcal{T})$, the reorganized sequences $(\mathbf{y}^{\text{BD}}, \mathcal{T}^{\text{BD}})$ have the same length, thus with no extra overhead.

We also adapt the self-attention mask to permit step-wise bidirectional generation:

$$\mathbf{M}^{\text{BD}} = \mathcal{M}(2, 1), \tag{8}$$

where IBDecoder has $h = 2$ generation directions. This corresponds to the relaxed causal mask by Wang et al. (2018), which ensures access to all predictions made in previous time steps[3] and allows for interactions among the tokens to be produced per time step. Although two words are predicted independently at each step, the adapted self-attention mask makes their corresponding decoding context complete; each word has full access to its corresponding decoding history, i.e. the left-to-right ($\mathbf{y}_{1:t}$) and right-to-left ($\mathbf{y}_{n-t+1:n}$) context. Except for $(\mathbf{y}^{\text{BD}}, \mathbf{M}^{\text{BD}}, \mathcal{T}^{\text{BD}})$, other components in Transformer are kept intact, including training objective.

4.1 Beyond Two-Word Generation

Eq. 2 only supports two-word generation, which indicates an upper bound of 2× speedup at inference. To improve this bound, we study strategies for multi-word generation. We explore two of them.

Multi-Directional Decoding Similar to IBDecoder, IMDecoder also permutes the target sequence. It inserts multiple generation directions (i.e. increases h), with each direction producing one word per step (i.e. $c = 1$). As shown in Figure 1d, it splits the target sequence into several roughly equal segments followed by applying IBDecoder to each segment (thus an even h required). Formally, IMDecoder reframes the target sequence and word positions as follows:

$$\mathbf{y}^{\text{MD}} = \left[\mathbf{y}_{1,k}^{\text{BD}}, \mathbf{y}_{2,k}^{\text{BD}}, \dots, \mathbf{y}_{h/2,k}^{\text{BD}}\right]_{k=1}^{\lceil n/h \rceil}, \tag{9}$$

$$\mathcal{T}_t^{\text{MD}} = \left(\lfloor {}^{t-1}/_{h} \rfloor, t - 1 \bmod h\right), \tag{10}$$

where $\mathbf{y}_{i,k}^{\text{BD}}$ denotes the k-th word of $\mathbf{y}_i^{\text{BD}}$, which is the i-th segment of $\mathbf{y}$ reordered by IBDecoder(${}^{h}/_{2}$ segments in total). $\mathcal{T}^{\text{MD}}$ decomposes the word position into two parts. The first one represents the index of decoding step where each word is predicted; the second one denotes the generation direction each target word belongs to. Specifically, we record the corresponding direction indices and add a group of trainable direction embeddings (red rectangles in Figure 1d) into the decoder input. IMDecoder uses the following self-attention mask:

$$\mathbf{M}^{\text{MD}} = \mathcal{M}(h, 1) \tag{11}$$

Semi-Autoregressive Decoding Instead of partitioning the target sequence, another option is to produce multiple target words per direction at each

[2] Consider Figure 1a. We cannot reorder position encodings along with embeddings (1,6,2,5,...) because we do not know sentence length at test time. Simply using vanilla position encodings (1,2,3,4,...) would increase the embedding distance between positions within a direction.

[3] Note that with two tokens produced per time step, decoder inputs are shifted by two.

Algorithm 1 Beam search with step-wise multi-word generation.

Input: Decoder *dec*, beam size B, word number $z = h \cdot c$, maximum length T
Output: Top-B finished hypothesis

▷ *initial hypothesis (z start symbols, score 0)*
1: $\mathcal{H}_0 \leftarrow \{([\text{`}[s]\text{'}]^z, 0)\}$
2: $\mathcal{H}_{finish} \leftarrow \emptyset$
3: $t \leftarrow 0$
4: **while** $|\mathcal{H}_{finish}| < B \ \ \& \ \ t < T$ **do**
5: **for** $(h_t, s_t) \in \mathcal{H}_t$ **do**
▷ *words W_p of probability $P \in \mathbb{R}^{z \times B}$*
6: $\mathbf{P}, \mathbf{W}_p \leftarrow top_B(dec(h_t))$
▷ *$\oplus$: outer addition for vectors*
7: $\mathbf{s}, \mathbf{W}_s \leftarrow top_B(\oplus_{i=1}^{z} \log \mathbf{P}_i)$
▷ *extract words by index, $W \in \mathbb{R}^{B \times z}$*
8: $\mathbf{W} \leftarrow tracewords(\mathbf{W}_s, \mathbf{W}_p)$
9: **for** $(\mathbf{w}, s)$ **in** $(\mathbf{W}, \mathbf{s})$ **do**
▷ *meet end-of-hypothesis condition*
10: **if** *finish*$(\mathbf{w})$ **then**
11: add $([h_t, \mathbf{w}], s + s_t)$ to $\mathcal{H}_{finish}$
12: **else**
13: add $([h_t, \mathbf{w}], s + s_t)$ to $\mathcal{H}_{t+z}$
14: **end if**
15: **end for**
16: **end for**
17: prune $\mathcal{H}_{t+z}$ to keep top-B hypothesis
18: $t \leftarrow t + z$
19: **end while**
▷ *post($\cdot$): process h_t to recover word order*
20: **return** sort $(post(h_t), s_t) \in \mathcal{H}_{finish}$ by $\frac{s_t}{t}$

step (i.e. increase c, Wang et al., 2018). SA assumes that neighbouring words are conditionally independent, despite the fact that tokens in natural language are typically highly inter-dependent.

We combine SA with IBDecoder (Figure 1e) with the expectation that producing 2 neighbouring tokens independently per direction is less harmful than producing 4 neighbouring words in parallel. We reuse the sequence $\mathbf{y}^{\text{BD}}$ and $\mathcal{T}^{\text{BD}}(n)$ for the decoder input, but enlarge the attention range in the self-attention mask to assist multi-word generation (Figure 1f):

$$\mathbf{M}^{\text{SA}} = \mathcal{M}(2, c). \tag{12}$$

4.2 Inference

To handle multiple predicted words per decoding step simultaneously, we adjust the beam search algorithm as in Algorithm 1. For each partial hypothesis h_t, we predict $z = h \cdot c$ words in parallel. We thus first extract the B top-scoring predictions $\mathbf{W}_p$ of probability $\mathbf{P}$ for all z positions (line 6), followed by pruning the resulting search space of size $\mathcal{O}(B^z)$ through an outer-addition operation to size B (line 7). The scores $\mathbf{s} \in \mathbb{R}^B$ (line 7) and the backtraced words $\mathbf{W} \in \mathbb{R}^{B \times z}$ (line 8) are then used for normal decoding. Note that each complete hypothesis requires a simple deterministic post-processing to recover its original word order (line 20). In contrast to Zhou et al. (2019), we do not separate the left-to-right beam from the right-to-left beam.

End-of-Hypothesis Condition With multiple predicted target words, determining whether one hypothesis is complete or not becomes challenging. We adopt a simple strategy: one hypothesis is assumed complete once any word in the predictions hits the end-of-sentence symbol ("[/s]") (line 10). We leave the study of alternatives for the future.

5 Experiments

Setup We test our model on machine translation (MT) and document summarization. We train MT models on five different language pairs: WMT14 English-German (En-De, Bojar et al., 2014), WMT14 English-French (En-Fr, Bojar et al., 2014), WMT16 Romanian-English (Ro-En, Bojar et al., 2016), WMT18 English-Russian (En-Ru, Bojar et al., 2018) and WAT17 Small-NMT English-Japanese (En-Ja, Nakazawa et al., 2017). Translation quality is measured by BLEU (Papineni et al., 2002), and we report detokenized BLEU using the toolkit *sacreBLEU* (Post, 2018)[4] except for En-Ja. Following Gu et al. (2019b), we segment Japanese text with KyTea[5] and compute tokenized BLEU. We train document summarization models on two benchmark datasets: the non-anonymized version of the CNN/Daily Mail dataset (CDMail, Hermann et al., 2015) and the Annotated English Gigaword (Gigaword, Rush et al., 2015). We evaluate the summarization quality using ROUGE-L (Lin, 2004).

We provide details of data preprocessing and model settings in Appendix A. We perform thorough analysis of our model on WMT14 En-De. We also report results improved by knowledge distillation (KD, Kim and Rush, 2016).

[4] Signature BLEU+c.mixed+#.1+s.exp+tok.13a+v.1.4.3
[5] http://www.phontron.com/kytea/

ID	Model	B	h	c	BLEU↑	+KD↑	Latency↓	Speedup↑	Train↑
1	Transformer	4 1	1	1	26.9 26.0	27.3 26.8	387 294	1.00× 1.32×	1.00×
2	IBDecoder	4 1	2	1	26.2 25.0	27.1 26.8	204 166	1.90× 2.33×	0.98×
3	2 + SA	4 1	2	2	23.0 21.7	26.3 26.0	117 89	3.31× 4.35×	0.98×
4	IMDecoder	4 1	4	1	21.5 19.7	24.6 24.1	102 85	3.79× 4.55×	0.98×

Table 1: Performance on WMT14 En-De for different models with respect to beam size (B), generation direction number (h, Eq. 4) and predicted token number per step (c, Eq. 4). *BLEU*: detokenized BLEU for models trained from scratch, *+KD*: detokenized BLEU for models trained with knowledge distillation. *Latency* (in millisecond) and *Speedup* are evaluated by decoding the test set with a batch size of 1, averaged over three runs. We report the latency and speedup for ②, ③ and ④ trained with KD. *Train* compares the training speed averaged over 100 steps. Time is measured on GeForce GTX 1080.

5.1 Results on WMT14 En-De

Table 1 compares the performance of our models on WMT14 En-De. Relaxing the autoregressiveness with IBDecoder yields slightly worse translation quality compared to Transformer (-0.7 BLEU, ①→②, w/o KD, $B = 4$). Unlike Zhang et al. (2020), we observe no quality improvement, but our model delivers a speedup of 1.90×∼2.33× at inference, clearly surpassing the simple greedy decoding baseline (1.32×) and BIFT (0.89×) (Zhang et al., 2020). The dropped quality is easily recovered with knowledge distillation (+0.2 BLEU, ①→②, w/ KD, $B = 4$).

Going beyond two-word generation, which enhances independence, greatly decreases the performance (②→③,④, w/o KD) while enlarging the speedup to 3.3×–4.5×. Compared to SA, the quality degradation with IMDecoder is larger, both w/ and w/o KD. We ascribe this to the difficulty of structure planning, as IMDecoder has to guess words in the middle of the sequence at the start of generation. We employ SA for the following experiments.

In contrast to existing work (Zhang et al., 2018d, 2019b, 2020; Zhou et al., 2019), our models marginally affect the training efficiency (0.98× vs 0.61× (Zhang et al., 2020)), and require no extra linguistic information (Akoury et al., 2019). Our results also suggest that the degree each model benefits from KD varies. Follow-up studies should report performance w/ and w/o KD.

Ablation Study We carry out an ablation study as shown in Table 2. Replacing the attention mask with the vanilla one (①→②) introduces unnecessary independence assumptions and reduces performance by 0.5 BLEU. Using vanilla positional encodings (③) also reduces performance -0.3 BLEU, indicating that we benefit from preserving the locality bias of sinusoidal encodings within each direction. Changing the generation direction from the side-to-middle (①) to the middle-to-side (④) dramatically increases the learning difficulty (-5.5 BLEU).

ID	Model	h	c	BLEU↑
1	IBDecoder	2	1	26.2
2	1 + vanilla mask	2	1	25.7
3	1 + vanilla positions	2	1	25.9
4	1 + middle-to-side	2	1	20.7
5	1 + indep. directions	2	1	23.9
6	vanilla SA	1	2	24.1
7	1 + SA	2	2	23.0
8	vanilla SA	1	4	18.7

Table 2: Ablation study on WMT14 En-De. Beam size 4. All models are trained from scratch. *vanilla mask*/*vanilla positions*: the self-attention mask ($\mathcal{M}(1, 1)$, Eq. 4) and word positions ($\mathcal{T}$, Eq. 5) used in Transformer. *middle-to-side*: generate words from the middle of the sequence to its two ends, a reverse mode of IBDecoder. *indep. directions*: disable cross-direction interaction. *vanilla SA*: predict multiple target words per step following one direction (Wang et al., 2018).

In IBDecoder, the two translation directions are interlinked, i.e. predictions are conditioned on the history of both directions. We can remove cross-direction attention, essentially forcing the model to produce the left and right half of sequences independently. Such an independent generation performs poorly (-2.3 BLEU, ①→⑤), which supports the importance of using bidirectional context and resonates with the finding of Zhou et al. (2019).

Vanilla SA vs. IBDecoder Our IBDecoder shares architectural properties with vanilla SA (Wang et al., 2018), namely the independent generation of two tokens per time step, and the

	Left-to-Right	Bidirectional
Autoregressive	4.04	4.86
Semi-Autoregressive	6.95	4.72
Estimated PMI	0.235	-0.014

Table 3: Perplexity of autoregressive and semi-autoregressive models with different factorizations, and estimated average point-wise mutual information between words that are predicted independently. Measured on WMT14 En-De test set. *Left-to-Right*: $h = 1$, *Bidirectional*: $h = 2$; Autoregressive: $z = 1$, Semi-autoregressive: $z = 2$. The estimated PMI shows that the inter-dependence of word pairs predicted in parallel by vanilla SA is stronger than for those predicted simultaneously by IBDecoder.

Model	$L/h/c$	BLEU↑	Speedup↑
Transformer	6/1/1	26.9	1.00×
+ student	2/1/1	26.0	2.19×
+ KD	2/1/1	26.7	2.32×
IBDecoder	6/2/1	26.2	1.90×
+ student	2/2/1	25.0	4.29×
+ KD	2/2/1	26.6	4.41×
IBDecoder + SA	6/2/2	23.0	3.31×
+ student	2/2/2	21.5	7.13×
+ KD	2/2/2	24.5	7.24×

Table 4: Detokenized BLEU and decoding speedup for student models on WMT14 En-De with reduced decoder depth L (encoder depth remains constant). Beam size 4.

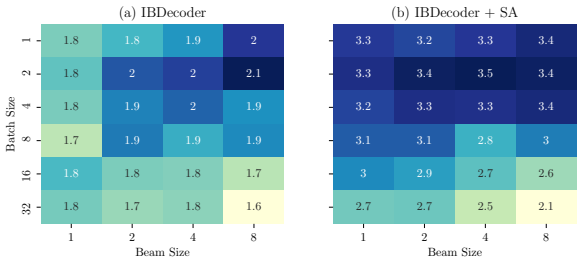


Figure 2: Speedup against Transformer vs. batch size and beam size on WMT14 En-De. Comparison is conducted under the same batch size and beam size. IBDecoder (+SA) is trained with KD. Our model consistently accelerates decoding.

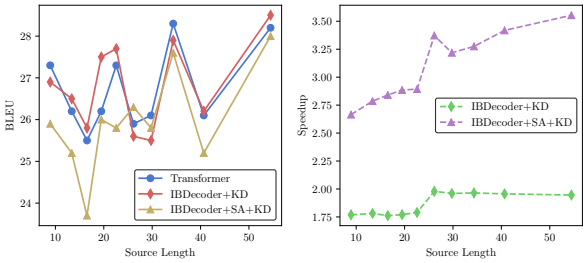


Figure 3: BLEU (solid lines, left) and speedup (dashed lines, right) as a function of source sentence length on WMT14 En-De. We sort the test set according to the source sentence length and uniformly divide it into 10 bins (274 sentences each). IBDecoder (+SA) is trained with KD. Beam size 4.

adapted self-attention mask, but crucially differ in their generation order and independence assumptions, with vanilla SA operating from left-to-right, and IBDecoder interleaving left-to-right and right-to-left decoding.

Our ablation results in Table 2 show that IBDecoder substantially outperforms vanilla SA (2.1/4.3 BLEU, ①→⑥/⑦→⑧). To further investigate the difference in independence assumptions between the two approaches, we compare estimated point-wise mutual information (PMI) of the words being predicted independently by IBDecoder and vanilla SA.[6] Results in Table 3 show that the PMI in IBDecoder (-0.014) is significantly smaller than that in vanilla SA (0.235), supporting our assumption that distant words are less inter-dependent on average. This also explains the smaller quality loss in IBDecoder compared to vanilla SA.

On Teacher-Student Model One classical approach to improving decoding efficiency is training a small student model w/ KD. Results in Table 4 support this: Transformer with a student model produces similar performance w/ KD but runs 2.32× faster, even better than IBDecoder (1.90×). Combining the student schema with IBDecoder increases the speedup to 4.41× without hurting the performance (26.6 BLEU, w/ KD). In exchange of 2.4 BLEU, we could reach 7.24× faster decoding with SA. The compatibility of our model with the teacher-student framework reflects the generalization of our bidirectional modeling. The results also demonstrate that efficiency improvements from faster autoregressive decoding, here obtained by reducing the number of decoder layers L[7], and from bidirectional decoding, are orthogonal.

Impact of Batch and Beam Size Figure 2 shows speedups over a standard Transformer with varying batch and beam sizes. When batch size < 8, increasing beam size improves the speedup; while the impact becomes negative with batch size ≥ 8. Overall, our model is consistently faster than Transformer at inference, regardless of the batch and beam size.

Impact of Source Sentence Length Although translation quality fluctuates over the source sentence length, Figure 3 shows that our model shares the same performance pattern with the baseline.

[6]Details about PMI estimation are given in Appendix B

[7]Also note the concurrent work by (Kasai et al., 2020).

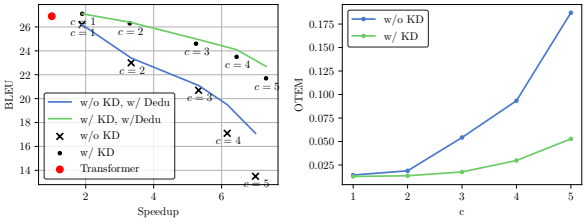


Figure 4: BLEU versus speedup (left) and OTEM (right) for different c on WMT14 En-De. Generation directions: $h = 2$. Beam size 4. OTEM↓: a metric measuring the degree of over-translation (Yang et al., 2018). Larger c indicates more independence between neighbouring tokens and results in more severe over-translation. Deduplication (Dedu) improves translation quality for large c.

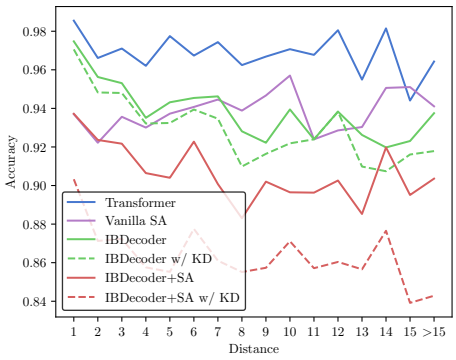


Figure 5: Accuracy of different models over distances on the subject-verb agreement task in *Lingeval97*.

With respect to the speedup, our model performs better when translating longer source sentences.

Effect of c Results in Figure 4 show that c controls the trade-off between translation quality and speedup. With larger c, more target tokens are predicted per decoding direction, leading to better speedup, but causing a larger performance drop w/ and w/o KD. Further analysis reveals that, as the dependency between predicted target words weakens, our model suffers from more serious over-translation issue, yielding larger OTEM (Yang et al., 2018). Although n-gram deduplication slightly improves quality[8], it does not explain the whole performance drop, echoing with Wang et al. (2018). We recommend using $c = 2$ for a good balance. In addition, the reduction of OTEM by KD in Figure 4 partially clarifies its improvement on quality.

Analysis on Long-range Dependency We adopt the subject-verb agreement task from *Lingeval97* (Sennrich, 2017) for analysis. We can see from the results in Figure 5 that IBDecoder performs similarly to the original Transformer for agreement over short distances, but agreement over longer distances drops on average. In contrast, models that include SA show steep drops in accuracy for short distances.

Curiously, KD seems to harm agreement scores even though it led to higher BLEU. Overall, these results suggest that BLEU does not show the full quality loss incurred by our independence assumptions. This deficiency also provides evidence for the performance drop in Figure 4.

[8] we only applied deduplication for results in Figure 4.

Model	BLEU↑	SU↑
Existing work		
SAT (Wang et al., 2018)*	26.09†	2.07 ×
SBSG (Zhou et al., 2019)*	27.22†	1.61 ×
SynST (Akoury et al., 2019)	20.74	4.86×
Levenshtein (Gu et al., 2019b)*	27.27†	4.01×
CMLM (Ghazvininejad et al., 2019)*	27.03†	-
AXE (Ghazvininejad et al., 2020)*	23.53†	-
This work	SacreBLEU↑	
IBDecoder	25.0	25.73† 2.48×
w/ SA	22.3◇	22.95† 4.53×
w/ student	25.0	25.33† 4.29×
IBDecoder*	26.8	27.50† 2.33×
w/ SA*	26.0◇	26.84†◇ 4.35×
w/ student*	26.6	27.00† 4.41×

Table 5: Comparison to several recent fast sequence generation models on WMT14 En-De. *: trained w/ KD. †: tokenized BLEU. ◇: deduplication applied. *SU*: short for speedup.

Comparison to Previous Work Results in Table 5 show that our model outperforms SynST (Akoury et al., 2019) in quality, and slightly surpasses the Levenshtein Transformer (Gu et al., 2019b) in speed. Particularly, our model ($27.50^{\dagger}/2.33\times$) surpasses SAT (Wang et al., 2018) ($26.09^{\dagger}/2.07\times$) and SBSG (Zhou et al., 2019) ($27.22^{\dagger}/1.61\times$) in terms of both quality and speed. Our model doesn't heavily rely on extra linguistic knowledge (Akoury et al., 2019), neither requires complex pseudo training data construction (Gu et al., 2019b). Compared to these prior studies, our approach is simple but effective.

5.2 Results on Other Tasks

Table 6 shows MT results for other translation directions, and for document summarization. Regardless of syntactic, morphological, transcript and sequence-length differences, our model achieves comparable generation quality and 1.75×–11.15× speedup over different tasks. With KD, our model even outperforms the Transformer baseline on 5 out of 6 tasks. In particular, our model succeeds on the

B		Model	KD	Machine Translation				Document Summarization	
				En-Fr	Ro-En	En-Ru	En-Ja	Gigaword	CDMail
4	Quality↑	Transformer	no	32.1	32.7	27.7	**43.97**	35.03	36.88
		IBDecoder	no	32.1	33.3	27.0	43.51	34.57	36.11
		+ SA	no	30.3	31.3	25.0	41.75	33.65	35.27
		IBDecoder	yes	**32.7**	**33.5**	27.5	43.76	35.12	36.46
		+ SA	yes	31.3	32.7	26.4	42.99	34.74	36.27
	Latency↓ /Speedup↑	IBDecoder	yes	231/1.75×	205/1.79×	204/1.82×	157/1.86×	83/2.35×	657/3.02×
		+SA	yes	119/3.41×	109/3.37×	112/3.30×	94/3.10×	47/4.20×	303/6.55×
1	Quality↑	Transformer	no	31.6	32.3	27.8	42.95	34.88	34.51
		IBDecoder	no	31.7	32.6	26.8	43.29	34.22	36.74
		+ SA	no	29.0	30.4	24.3	41.05	33.25	35.04
		IBDecoder	yes	32.2	33.2	**28.2**	43.79	**35.18**	**37.03**
		+ SA	yes	30.7	32.4	26.5	42.70	34.63	36.39
	Latency↓ /Speedup↑	Transformer	no	357/1.14×	333/1.10×	342/1.09×	260/1.12×	157/1.24×	1447/1.37×
		IBDecoder	yes	186/2.18×	154/2.37×	157/2.37×	121/2.40×	56/3.51×	312/6.36×
		+SA	yes	96/4.20×	88/4.17×	90/4.14×	67/4.34×	34/5.83×	178/11.15×

Table 6: Generation quality (BLEU for MT, Rouge-L for summarization) and latency(ms)/speedup on different tasks. We compare IBDecoder (+SA) with Transformer. Best quality is in **bold**.

CDMail task which previous non-autoregressive models rarely attempt due to its lengthy target sequence, although our model suffers from the long-range dependency issue as in Figure 5.

6 Conclusion and Future Work

We present interleaved bidirectional sequence generation to accelerate decoding by enabling generation from the left-to-right and right-to-left directions simultaneously. We combine the strengths of SBSG (Zhou et al., 2019) and SA (Wang et al., 2018), and propose a simple interleaved bidirectional decoder (IBDecoder) that can be easily implemented on top of a standard unidirectional decoder, like Transformer, via interleaving the target sequence and tweaking the word positions and self-attention masks. IBDecoder inherits Transformer's training parallelization with no additional model parameters, and is extensible with SA and multi-directional decoding. We show that the independence assumptions we introduce between the two directions are less harmful to translation quality than the independence assumptions in left-to-right SA. On a series of generation tasks, we report comparable quality with significant inference speedup (4×–11×) and little training overhead. We also show that the approach is orthogonal to speedups to autoregressive decoding, e.g. by reducing model size.

In the future, we would like to further improve multi-directional generation, and will investigate alternative ways to partition the target sequence and encode positional information. We are also interested in better measuring and reducing the quality loss resulting from long-distance dependencies. Finally, we would like to adapt our interleaving approach to other sequence-to-sequence architectures.

Acknowledgments

This work was performed using resources provided by the Cambridge Service for Data Driven Discovery (CSD3) operated by the University of Cambridge Research Computing Service (http://www.csd3.cam.ac.uk/), provided by Dell EMC and Intel using Tier-2 funding from the Engineering and Physical Sciences Research Council (capital grant EP/P020259/1), and DiRAC funding from the Science and Technology Facilities Council (www.dirac.ac.uk). Ivan Titov acknowledges support of the European Research Council (ERC Starting grant 678254) and the Dutch National Science Foundation (NWO VIDI 639.022.518). Rico Sennrich acknowledges support of the Swiss National Science Foundation (MUTAMUR; no. 176727).

References

Nader Akoury, Kalpesh Krishna, and Mohit Iyyer. 2019. Syntactically supervised transformers for faster neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1269–1281, Florence, Italy. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly

learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Aishwarya Bhandare, Vamsi Sripathi, Deepthi Karkada, Vivek Menon, Sun Choi, Kushal Datta, and Vikram Saletore. 2019. Efficient 8-bit quantization of transformer neural machine language translation model. *arXiv preprint arXiv:1906.00532*.

Ondrej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Aleš Tamchyna. 2014. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 12–58, Baltimore, Maryland, USA. Association for Computational Linguistics.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurélie Névéol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. 2016. Findings of the 2016 conference on machine translation. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 131–198, Berlin, Germany. Association for Computational Linguistics.

Ondřej Bojar, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Philipp Koehn, and Christof Monz. 2018. Findings of the 2018 conference on machine translation (WMT18). In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 272–303, Belgium, Brussels. Association for Computational Linguistics.

Dmitrii Emelianenko, Elena Voita, and Pavel Serdyukov. 2019. Sequence modeling with unconstrained generation order. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 7700–7711. Curran Associates, Inc.

Andrew Finch and Eiichiro Sumita. 2009. Bidirectional phrase-based statistical machine translation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1124–1132, Singapore. Association for Computational Linguistics.

Marjan Ghazvininejad, Vladimir Karpukhin, Luke Zettlemoyer, and Omer Levy. 2020. Aligned cross entropy for non-autoregressive machine translation. *ArXiv*, abs/2004.01655.

Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. 2019. Mask-predict: Parallel decoding of conditional masked language models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6112–6121, Hong Kong, China. Association for Computational Linguistics.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor O.K. Li, and Richard Socher. 2018. Non-autoregressive neural machine translation. In *International Conference on Learning Representations*.

Jiatao Gu, Qi Liu, and Kyunghyun Cho. 2019a. Insertion-based decoding with automatically inferred generation order. *Transactions of the Association for Computational Linguistics*, 7:661–676.

Jiatao Gu, Changhan Wang, and Junbo Zhao. 2019b. Levenshtein transformer. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 11181–11191. Curran Associates, Inc.

Junliang Guo, Xu Tan, Di He, Tao Qin, Linli Xu, and Tie-Yan Liu. 2019. Non-autoregressive neural machine translation with enhanced decoder input. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3723–3730.

Hiroaki Hayashi, Yusuke Oda, Alexandra Birch, Ioannis Konstas, Andrew Finch, Minh-Thang Luong, Graham Neubig, and Katsuhito Sudoh. 2019. Findings of the third workshop on neural generation and translation. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 1–14, Hong Kong. Association for Computational Linguistics.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 1693–1701. Curran Associates, Inc.

Xiaoguang Hu, Wei Li, Xiang Lan, Hua Wu, and Haifeng Wang. 2015. Improved beam search with constrained softmax for nmt. *Proceedings of MT Summit XV*, page 297.

Lukasz Kaiser, Samy Bengio, Aurko Roy, Ashish Vaswani, Niki Parmar, Jakob Uszkoreit, and Noam Shazeer. 2018. Fast decoding in sequence models using discrete latent variables. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2390–2399, Stockholmsmässan, Stockholm Sweden. PMLR.

Shigeki Karita, Nanxin Chen, Tomoki Hayashi, Takaaki Hori, Hirofumi Inaguma, Ziyan Jiang, Masao Someki, Nelson Enrique Yalta Soplin, Ryuichi Yamamoto, Xiaofei Wang, Shinji Watanabe,

Takenori Yoshimura, and Wangyou Zhang. 2019. A comparative study on transformer vs rnn in speech applications. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 449–456.

Jungo Kasai, Nikolaos Pappas, Hao Peng, James Cross, and Noah A. Smith. 2020. Deep encoder, shallow decoder: Reevaluating the speed-quality tradeoff in machine translation.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327, Austin, Texas. Association for Computational Linguistics.

Young Jin Kim, Marcin Junczys-Dowmunt, Hany Hassan, Alham Fikri Aji, Kenneth Heafield, Roman Grundkiewicz, and Nikolay Bogoychev. 2019. From research to production and back: Ludicrously fast neural machine translation. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 280–288, Hong Kong. Association for Computational Linguistics.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*.

Jason Lee, Elman Mansimov, and Kyunghyun Cho. 2018. Deterministic non-autoregressive neural sequence modeling by iterative refinement. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1173–1182, Brussels, Belgium. Association for Computational Linguistics.

Jindřich Libovický and Jindřich Helcl. 2018. End-to-end non-autoregressive neural machine translation with connectionist temporal classification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3016–3021, Brussels, Belgium. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Lemao Liu, Masao Utiyama, Andrew Finch, and Eiichiro Sumita. 2016. Agreement on target-bidirectional neural machine translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 411–416, San Diego, California. Association for Computational Linguistics.

Toshiaki Nakazawa, Shohei Higashiyama, Chenchen Ding, Hideya Mino, Isao Goto, Hideto Kazawa, Yusuke Oda, Graham Neubig, and Sadao Kurohashi. 2017. Overview of the 4th workshop on Asian translation. In *Proceedings of the 4th Workshop on Asian Translation (WAT2017)*, pages 1–54, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv e-prints*.

Qiu Ran, Yankai Lin, Peng Li, and Jie Zhou. 2020. Learning to recover from multi-modality errors for non-autoregressive neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3059–3069, Online. Association for Computational Linguistics.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389, Lisbon, Portugal. Association for Computational Linguistics.

Abigail See, Minh-Thang Luong, and Christopher D. Manning. 2016. Compression of neural machine translation models via pruning. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 291–301, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich. 2017. How grammatical is character-level neural machine translation? assessing MT quality with contrastive translation pairs. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 376–382, Valencia, Spain. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Edinburgh neural machine translation systems for WMT 16. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 371–376, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational*

Linguistics (Volume 1: Long Papers), pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Chenze Shao, Jinchao Zhang, Yang Feng, Fandong Meng, and Jie Zhou. 2020. Minimizing the bag-of-ngrams difference for non-autoregressive neural machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Noam Shazeer. 2019. Fast transformer decoding: One write-head is all you need. *arXiv preprint arXiv:1911.02150*.

Linfeng Song, Daniel Gildea, Yue Zhang, Zhiguo Wang, and Jinsong Su. 2019. Semantic neural machine translation using amr. *Transactions of the Association for Computational Linguistics*, 7:19–31.

Mitchell Stern, William Chan, Jamie Kiros, and Jakob Uszkoreit. 2019. Insertion transformer: Flexible sequence generation via insertion operations. In *International Conference on Machine Learning*, pages 5976–5985.

Mitchell Stern, Noam Shazeer, and Jakob Uszkoreit. 2018. Blockwise parallel decoding for deep autoregressive models. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 10086–10095. Curran Associates, Inc.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Chengyi Wang, Shuangzhi Wu, and Shujie Liu. 2019. Accelerating transformer decoding via a hybrid of self-attention and recurrent neural network. *arXiv preprint arXiv:1909.02279*.

Chunqi Wang, Ji Zhang, and Haiqing Chen. 2018. Semi-autoregressive neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 479–488, Brussels, Belgium. Association for Computational Linguistics.

Taro Watanabe and Eiichiro Sumita. 2002. Bidirectional decoding for statistical machine translation. In *COLING 2002: The 19th International Conference on Computational Linguistics*.

Felix Wu, Angela Fan, Alexei Baevski, Yann Dauphin, and Michael Auli. 2019. Pay less attention with lightweight and dynamic convolutions. In *International Conference on Learning Representations*.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.

Jing Yang, Biao Zhang, Yue Qin, Xiangwen Zhang, Qian Lin, and Jinsong Su. 2018. Otem&utem: Over- and under-translation evaluation metric for nmt. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 291–302. Springer.

Biao Zhang and Rico Sennrich. 2019. A lightweight recurrent network for sequence modeling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1538–1548, Florence, Italy. Association for Computational Linguistics.

Biao Zhang, Ivan Titov, and Rico Sennrich. 2019a. Improving deep transformer with depth-scaled initialization and merged attention. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 898–909, Hong Kong, China. Association for Computational Linguistics.

Biao Zhang, Deyi Xiong, and Jinsong Su. 2018a. Accelerating neural transformer via an average attention network. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1789–1798, Melbourne, Australia. Association for Computational Linguistics.

Biao Zhang, Deyi Xiong, Jinsong Su, Qian Lin, and Huiji Zhang. 2018b. Simplifying neural machine translation with addition-subtraction twin-gated recurrent networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4273–4283, Brussels, Belgium. Association for Computational Linguistics.

Biao Zhang, Deyi Xiong, Jinsong Su, and Jiebo Luo. 2019b. Future-aware knowledge distillation for neural machine translation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27(12):2278–2287.

Jiajun Zhang, Long Zhou, Yang Zhao, and Chengqing Zong. 2020. Synchronous bidirectional inference for neural sequence generation. *Artificial Intelligence*, 281:103234.

Wen Zhang, Liang Huang, Yang Feng, Lei Shen, and Qun Liu. 2018c. Speeding up neural machine translation decoding by cube pruning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4284–4294, Brussels, Belgium. Association for Computational Linguistics.

Xiangwen Zhang, Jinsong Su, Yue Qin, Yang Liu, Rongrong Ji, and Hongji Wang. 2018d. Asynchronous bidirectional decoding for neural machine translation. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Zhirui Zhang, Shuangzhi Wu, Shujie Liu, Mu Li, Ming Zhou, and Tong Xu. 2019c. Regularizing neural machine translation by target-bidirectional agreement. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 443–450.

Zaixiang Zheng, Shujian Huang, Zhaopeng Tu, Xin-Yu Dai, and Jiajun Chen. 2019. Dynamic past and future for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 931–941, Hong Kong, China. Association for Computational Linguistics.

Long Zhou, Jiajun Zhang, Chengqing Zong, and Heng Yu. 2019. Sequence generation: from both sides to the middle. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 5471–5477. AAAI Press.

A Data Preprocessing and Model Settings

We use the given well-processed data for WAT17 En-Ja. For other tasks, we apply the byte pair encoding model (Sennrich et al., 2016b) with a joint vocab size of 32K except for WMT18 En-Ru (48K). We experiment with Transformer Base (Vaswani et al., 2017): $d = 512$, $L = 6$, 8 attention heads and FFN size of 2048. Dropout of rate 0.1 is used on residual connections and attention weights. We employ Adam ($\beta_1 = 0.9, \beta_2 = 0.98$) (Kingma and Ba, 2015) for parameter optimization with a scheduled learning rate of warm-up step 4K. Gradient is estimated over roughly 25K target subwords. We average the last 5 checkpoints for evaluation, and use beam search (beam size 4, length penalty 0.6) by default for inference.

B Estimation of the PMI

To evaluate the average point-wise mutual information (PMI) in Table 3, we compare IBDecoder/vanilla SA to its autoregressive counterpart in terms of testing perplexity (ppl). Take SA ($h = 1, c = 2$) as example, we have:

$$\text{PMI}(SA) = \log \text{ppl}(SA) - \log \text{ppl}(\text{Base}) \quad (13)$$

where *Base* denotes the baseline Transformer. The intuition behind our estimation is that Transformer handles neighboring words (y_1, y_2) autoregressively, thus models their joint probability: $p(y_1, y_2) = p(y_1) \cdot p(y_2|y_1)$. Instead, vanilla SA predicts those words independently, i.e. $p(y_1) \cdot p(y_2)$. Comparing the perplexity of SA and Transformer gives an estimation of the average PMI. The method for IBDecoder follows the same spirit.

Priming Neural Machine Translation

MinhQuang Pham[†‡], **Jitao Xu**[†‡], **Josep Crego**[†], **Jean Senellart**[†], **François Yvon**[‡]

[†]SYSTRAN, 5 rue Feydeau, 75002 Paris, France
firstname.lastname@systrangroup.com
[‡]Université Paris-Saclay, CNRS, LIMSI, 91400, Orsay, France
firstname.lastname@limsi.fr

Abstract

Priming is a well known and studied psychology phenomenon based on the prior presentation of one stimulus (cue) to influence the processing of a response. In this paper, we propose a framework to mimic the process of priming in the context of neural machine translation (NMT). We evaluate the effect of using similar translations as priming cues on the NMT network. We propose a method to inject priming cues into the NMT network and compare our framework to other mechanisms that perform micro-adaptation during inference. Overall, experiments conducted in a multi-domain setting confirm that adding priming cues in the NMT decoder can go a long way towards improving the translation accuracy. Besides, we show the suitability of our framework to gather valuable information for an NMT network from monolingual resources.

1 Introduction

Priming is a well studied human cognitive phenomenon, founded on the establishment of associations between a stimulus and a response (Tulving et al., 1982). Multiple studies have shown how external stimuli (cues) may have a profound effect on perception. In the case of language translation, external stimuli having such effects are said to prime language understanding and potentially have a impact the actions of a human translator. Imagine for instance a translator facing the ambiguous sentence *I was in the bank*, and the effect on translation accuracy if primed with the cue *river*. Most likely, the human translator would consider the "edge of river" sense rather than "financial institution" for translation. In the context of human translation, cross-lingual priming is particularly effective as cues in the target language may notably influence the final translation word choice.

Several research works have introduced the priming analogy in deep neural networks. In computer vision priming has been broadly studied: for instance, in Rosenfeld et al. (2018), the authors introduce a cue about the presence of a certain class of object in an image that significantly improves object detection performance. Concerning language generation, Brown et al. (2020) use a combination of prompt and example to guide the GPT-3 network when performing a task, where the prompt is a sentence that describes the task (*i.e. "Translate from English to French"*); and is followed by an example of the task (*i.e. "sea otter ⇝ loutre de mer"*). In the context of NMT, experiments reported (Sennrich et al., 2016a; Kobus et al., 2017; Dinu et al., 2019) aim at influencing translation inference with respectively politeness, domain and terminology constraints. More related to our work, (Bulte and Tezcan, 2019; Xu et al., 2020) introduce a simple and elegant framework where similar translations (cues) are used to prime an NMT model, effectively boosting translation accuracy. In all cases, priming is performed by injecting cues in the input stream prior to inference decoding.

In this paper, we extend a framework that mimics the priming process in neural networks, in the context of machine translation. Following up on previous work (Bulte and Tezcan, 2019; Xu et al., 2020), we consider similar translations as external cues that can influence the translation process. We push this concept further: a) by proposing a novel scheme to integrate similar translation cues into the NMT network. We examine the attention mechanism of the network and confirm that priming stimuli are actually taken into account; b) by extending an efficient network to train distributed representations of sentences that are used to identify accurate translations used as priming cues[1]; c) by analyzing how on-the-fly priming compares to micro-adaptation (fine-tuning). Finally, we

[1] https://github.com/jmcrego/cbon

Proceedings of the 5th Conference on Machine Translation (WMT), pages 516–527
Online, November 19–20, 2020.

show that our priming approach can also be used with monolingual data, providing a scenario where NMT can be effectively helped by large amounts of available data. Our proposal does not require to change the NMT architectures or algorithms, relying solely on input preprocessing and on prefix (forced) decoding (Santy et al., 2019; Knowles and Koehn, 2016), a feature already implemented in many NMT toolkits.

The remainder of the paper is organized as follows: Section 2 gives details regarding our priming approach. The experimental framework is presented in Section 3. Results and discussion are respectively in Sections 4 and 5. We review related work in Section 6 and conclude in Section 7.

2 NMT Priming On-the-fly

This section describes our framework for priming neural MT with similar translations. We follow the work by (Bulte and Tezcan, 2019; Xu et al., 2020) and build a translation model that incorporates similar translations from a translation memory (TM) to boost translation accuracy. In this work, TMs are parallel corpora containing translations falling in the same domain as test sentences.

We first describe the methods employed in this work to compute sentence similarity. We then introduce various augmentation schemes considered to prime the NMT network with retrieved similar translations. Overall, we pay special attention to efficiency, since retrieval is applied on a sentence-by-sentence basis at inference.

2.1 Similarity Computation

We detail the sentence similarity tools evaluated in this work. The first employs discrete word representations, while the rest rely on building distributed representations of sentences to perform similar sentence retrieval:

FM: fuzzy matching is a lexicalized matching method aimed to identify non-exact matches of a given sentence. Following Xu et al. (2020), we use `FuzzyMatch`[2], where the fuzzy match score $\mathrm{FM}(s_i, s_j)$ between two sentences s_i and s_j is:

$$\mathrm{FM}(s_i, s_j) = 1 - \frac{\mathrm{ED}(s_i, s_j)}{max(|s_i|, |s_j|)}$$

with $\mathrm{ED}(s_i, s_j)$ being the Edit Distance between s_i and s_j, and $|s|$ is the length of s.

S2V: we use `sent2vec`[3] (Pagliardini et al., 2018) to generate sentence embeddings. The network implements a simple but efficient unsupervised objective to train distributed representations for sentences. The model is based on efficient matrix factor (bilinear) models (Mikolov et al., 2013a,b; Pennington et al., 2014).

Borrowing the notations of Pagliardini et al. (2018), training the model is formalized as an optimization problem:

$$\min_{\boldsymbol{U},\boldsymbol{V}} \sum_{s \in \mathcal{C}} f_s(\boldsymbol{U}\boldsymbol{V}\iota_s)$$

for two parameter matrices $\boldsymbol{U} \in \mathbb{R}^{|\mathcal{V}| \times d}$ and $\boldsymbol{V} \in \mathbb{R}^{d \times |\mathcal{V}|}$, where $\mathcal{V}$ denotes the vocabulary and d is the embedding dimension. Minimization of the cost function f_s is performed on a training corpus $\mathcal{C}$ of sentences s.

In `sent2vec`, ι_s is a binary vector encoding the bigrams in s (bag of bigrams encoding).

CBON: the Continuous Bag of n-grams (`CBON`) model denotes our re-implementation of the previous `sent2vec` model. In addition to multiple implementation details, the main difference is the use of arbitrary large n-grams to model sentence representations, where `sent2vec` only used bigrams.

Both `sent2vec` and `CBON` learn a source (or context) embedding $\boldsymbol{v}_w$ for each n-gram w in the vocabulary $\mathcal{V}$. Once the model is trained, the embedding of sentence s (h_s) is obtained as the average of its n-gram embeddings:

$$\boldsymbol{h}_s = \frac{1}{|R(s)|} \sum_{w \in R(s)} \boldsymbol{v}_w$$

where $R(s)$ is the list of n-grams (including unigrams) occuring in sentence s and $\boldsymbol{v}_w$ is the target embedding of n-gram w.

The similarity score $\mathrm{EM}(s_i, s_j)$ between two sentences s_i and s_j is then defined via the cosine similarity of their sentence vector representations h_i and h_j:

$$\mathrm{EM}(s_i, s_j) = \frac{h_i \cdot h_j}{||h_i|| \times ||h_j||},$$

where $||h||$ denotes the norm of vector h.

Note that models differ in their vocabularies, which are built selecting the most frequent n-grams.

[2] https://github.com/systran/FuzzyMatch

[3] https://github.com/epfml/sent2vec

Both models implement Negative Sampling to avoid the softmax computation.

2.2 Priming Schemes

We now explore various ways to integrate similar translations for priming NMT:

tgt^k we follow here mostly the work of Bulte and Tezcan (2019), where the input sentence in the source language is augmented with the k translations (in the target language) having the highest matching score (FM or EM) in the TM.

In training, sentence pairs (**s**,**t**) are preprocessed as follows: the source sentence **s** is concatenated with translations t^k of the k most similar sentences (s^k) to **s** found in the TM. Augmented translations are sorted by matching score, with $k = 1$ denoting the most similar. Sentences in the source stream are separated using the special token ∘.

$$\texttt{src:} \quad t^k \circ ... \circ t^2 \circ t^1 \circ \mathbf{s}$$
$$\texttt{tgt:} \quad \mathbf{t}$$

In inference, only the source-side is input to the translation network.

In Xu et al. (2020), an issue regarding *unrelated* tokens present in similar translations t^k is raised. The model effectively learns to copy most of the content present in similar translations, but has difficulties to avoid also copying *unrelated* words. Consider for instance the input sentence $s =$ *pertussis vaccin* with similar sentence $s^1 =$ *measles vaccin* and its corresponding translation $t^1 =$ *vaccin contre la rougeole*. Following the **tgt**^{k} scheme, the NMT input consists of:

vaccin contre la rougeole ∘ **pertussis vaccin**

yielding the output: **vaccin contre la rougeole**. The word *rougeole* is actually the translation of an unrelated word (*measles*). The model often copies such *unrelated* tokens (Xu et al., 2020), due to the fact that they are present in the input stream as similar translations (t^k) and are usually semantically related to the correct translation choice (here *coqueluche*, the correct translation for *pertussis*).

tgt^k**+STU** adopts the proposal of Xu et al. (2020) to alleviate the *unrelated word* problem. It relies on an additional source stream (factor) to label related/unrelated tokens. Following on our example, in this scheme the input of the NMT model contains two parallel streams:

src$_1$:	vaccin	contre	la	rugeole	∘	**pertussis**	**vaccin**
src$_2$:	T	T	T	U	T	S	S
tgt:	vaccin	contre	la	coqueluche			

Tokens in the second stream are: S for source tokens, U for unrelated and T for related target tokens. *rougeole* is thus tagged as an *unrelated* word that must not be copied in the translation output. Word embeddings are built after concatenating both factor embeddings. Xu et al. (2020) claim achieving a 8% reduction of unrelated tokens when using this scheme.

Note that this solution is computationally expensive as it requires to identify related/unrelated tokens in each input sentence and in the corresponding similar translations, based in Xu et al. (2020) on word alignments and edit distance computations.

s+t^k the solution proposed in this paper also addresses the *unrelated word* problem, at a much reduced computational cost. It considers both sides of similar translations (s^k and t^k). Training streams take the form:

$$\texttt{src:} \quad s^k \circ ... \circ s^2 \circ s^1 \circ \mathbf{s}$$
$$\texttt{tgt:} \quad t^k \circ ... \circ t^2 \circ t^1 \circ \mathbf{t}$$

In inference, target-side similar translations t^k are used by the model as a target prefix. The initial steps of the beam search use the given prefix $t^k \circ ... \circ t^2 \circ t^1 \circ$ in forced decoding mode, returning to a regular beam search after the last ∘ token is generated. A similar strategy of concatenating previous and current sentences was explored by Tiedemann and Scherrer (2017) in the context of handling discourse phenomena. However, since we use true translation as prefixes, our strategy does not suffer from exposure bias (Ranzato et al., 2016) and the subsequent error propagation problem. Continuing on our running example, during inference the model receives:

$$\texttt{input:} \quad \textit{measles vaccin} \circ \mathbf{pertussis\ vaccin}$$
$$\texttt{prefix:} \quad \textit{vaccin contre la rougeole} \circ$$

the encoder embeds the input stream, and force-decodes the target prefix, before starting the translation generation. Note that during beam search, the decoder has thus access both to all input tokens (s^k and s) as well as to similar translations t^k (in the translation prefix).

Following our approach the NMT model learns to attend to priming cues on both source and target streams. Besides, our solution removes the need to mix source and target vocabularies as in previous schemes.

3 Experimental Framework

3.1 Corpora

We experiment with the English-French language pair and data originating from eight domains, corresponding to texts from three European institutions: the European Parliament (EPPS), the European Medicines Agency (EMEA) and the European Central Bank (ECB); Legislative texts of the European Union (JRC); IT-domain corpora corresponding to KDE4 and GNOME; News Commentaries (NEWS); and parallel sentences extracted from Wikipedia (WIKI). Table 1 contains statistics regarding the corpora used in this work[4] (Tiedemann, 2012). Statistics are computed after splitting off punctuation.

Corpus	#Sents (K)	L_{mean}		Vocab (K)	
		English	French	English	French
Parallel Corpora					
EPPS	1,992.8	27.7	32.0	129.5	149.2
NEWS	315.3	25.3	31.7	90.5	96.7
WIKI	749.0	25.9	23.5	527.5	506.6
ECB	174.1	28.6	33.8	45.3	53.5
EMEA	336.8	16.8	20.3	62.8	68.9
JRC	475.2	30.1	34.5	81.0	83.5
GNOME	51.9	9.6	11.6	19.0	21.6
KDE4	163.9	9.1	12.4	48.7	64.7
Monolingual Corpora					
WIKI	6,426.8	-	24.1	-	1,626.3
NEWS	83,567.8	-	25.5	-	3,444.1

Table 1: Corpora statistics. Note that K stands for thousands and L_{mean} is the average length in words.

Each corpora is considered as a different domain. Training data sets are also employed as TM of the corresponding domain. This is, similar sentences are mined from the same training set that is used to build the model. Note that we also consider monolingual (French) corpora. For the News domain we use all available monolingual WMT news crawl data[5]. For the Wikipedia domain, we use the French-side of the WikiMatrix data (Schwenk et al., 2019a).

We randomly split the parallel corpora by keeping 500 sentences for validation, 1,000 sentences for testing and the rest for training. All data is preprocessed using the OpenNMT tokenizer[6] (conservative mode).

[4]Freely available from http://opus.nlpl.eu

[5]http://data.statmt.org/news-crawl/

[6]https://github.com/OpenNMT/Tokenizer

3.2 System Configurations

This section gives learning/inference details of the various systems used in this work.

Similarity

For fuzzy matching **FM** we follow several works (Koehn and Senellart, 2010; Bulte and Tezcan, 2019; Xu et al., 2020) and keep the n-best matches when $\mathrm{FM}(s_1, s_2) \geq 0.5$ with no approximation. Concerning **S2V**, the model is trained with default options during 20 epochs using all training data. We use an embedding dimension of 300 cells. Regarding **CBON**, we learn models using also the entire training data during one epoch (∼50,000 iterations). Similarly to **S2V** we use 10 negative samples per positive word to approximate the softmax, a batch size of $2k$ examples, and embedding size of 300 cells. We build CBON models using 3-grams and 4-grams to enable a comparison with sent2vec which only uses bigrams. All vocabularies are selected keeping the 500,000 most frequent n-grams ($n = 2$ for **S2V** and $n = 3$ and 4 for **CBON**).

For both **CBON** and **S2V** models, we use the 5-best matches when $\mathrm{EM}(s_1, s_2) \geq 0.8$[7]. In all cases, perfect matches are not used for training. Accuracy results on the priming task indicate that 3-grams yield slightly lower accuracy results than those obtained with 4-grams. In the remainder, we always use the 4-gram version of **CBON**.

Sentence Retrieval

To identify similar translations using distributed representations, we use the faiss[8] search toolkit (Johnson et al., 2019) through its Python API with exact *FlatIP* index.

Translation

Our NMT models rely on the Transformer base architecture of Vaswani et al. (2017), implemented in the OpenNMT-tf[9] toolkit (Klein et al., 2017). We use the standard setting of Transformers for all experiments: size of word embedding: 512; size of hidden layers: 512; size of inner feed-forward layer: 2,048; number of heads: 8; number of layers in the encoder or in the decoder: 6. In the **tgt**[1]**+STU** scheme, token (508 cells) and **STU** (4

[7]Optimization experiments on a held-out development set are carried out for both models.

[8]https://github.com/facebookresearch/faiss

[9]https://github.com/OpenNMT/OpenNMT-tf

cells) streams are concatenated, thus using the same number of parameters in all schemes.

For training, we use the Adam (Kingma and Ba, 2015) optimiser with a batch size of $4,096$ tokens. We set the warmup steps to $4,000$ and update the learning rate for every 8 iterations. Models are optimised during $300K$ iterations, using a single NVIDIA V100 GPU. We limit the length of training sentences to 300 BPE tokens (Sennrich et al., 2016c) in both source and target sides to enable the integration of similar sentences. We use a joint BPE-vocabulary of size 32K for both source and target texts. Inference is performed with a beam size of 5 using CTranslate2[10], a custom C++ runtime inference engine for OpenNMT models that enables fast CPU decoding and also implements prefix decoding. For evaluation, we report BLEU (Papineni et al., 2002) scores computed by detokenized case-sensitive `multi-bleu.perl`[11].

We re-implement the work of Farajian et al. (2017) as a contrastive model that we denote μ**adapt**. Note that we only experiment with the basic version of this work, where the closest neighbours of the input sentence are first retrieved from the memory and then used to fine-tune a generic model during 15 additional iterations with a fixed learning rate of 0.0005; the fine-tuned model is then used to produce the translation of the given input sentence. In addition, Farajian et al. (2017) include a variant where learning rate and number of epochs are dynamically adapted considering sentence similarity. Adaptation is run on a sentence-by-sentence basis.

4 Results

Retrieval algorithms employed in this work are significantly faster than NMT Transformer decoding, thus implying a limited decoding overhead.

Table 2 reports efficiency scores (tokens/second) for computing vector representations (Vector), performing sentence retrieval (Retrieval) and translation (NMT) for the `WIKI` test set according to the similarity model and priming schema used. Results show that the computational cost is dominated by the NMT step. This step, in turn, is affected by the length of the input (and prefix) streams.

[10] https://github.com/OpenNMT/CTranslate2

[11] https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/multi-bleu.perl

Model	Schema	Vector	Retrieval	NMT
Base	-	-	-	806
FM	**tgt**^{1}	-	25K	750
	s+t^1			687
S2V	**tgt**^{5}	222K	17K	639
CBON	**tgt**^{5}	59K		
	s+t^5			523

Table 2: Efficiency (tokens/second) of each step for different inference configurations. All steps run on CPU (16 cores). K stands for thousands.

Table 3 reports `BLEU` scores for our various configurations, tested on 8 domain-specific test sets. The last column (avg) reports average results. This table also reports the number of input sentences (out of $1,000$) for which at least one similar sentence was retrieved (in a smaller font).

All NMT models are built using the concatenation of the original parallel corpora in Table 1. Our **Base** configuration does not integrate similar sentences in the training data. All other models extend the original corpora with sentences retrieved following similarity methods (Sim) introduced in Section 2.1 and integration schemes presented in Section 2.2 (Scheme).

The second block of results in Table 3 displays scores obtained when performing translations extended with fuzzy matches **FM**. In line with results presented by Xu et al. (2020), using a second stream to mark related/unrelated tokens (**+STU**) yields a boost in performance of around 1 `BLEU` points. When the **s+t**^{1} scheme is used, the average improvement reaches 1.25 `BLEU` points.

The third block compares translation results obtained when identifying similar translations by **S2V** and **CBON**. In both cases, the **s+t**^{5} scheme is used. The choice for 5-best similar translations and $\mathrm{EM}(s_i, s_j) \geq 0.8$ threshold is made after running optimization work on a held out development set. Sentences identified by **CBON** outperform those selected by **S2V**. The idiosyncrasy of fuzzy matching does not enable to find multiple similar sentences for a given input sentence. Overall best results are obtained by the **CBON s+t**^{5} configuration. Note that as expected, the number of similar translations found using distributed representations is larger than those found by fuzzy matching.

Finally, the last block in Table 3 gives results for a system that retrieves similar sentences to dynamically adapt the model on a sentence-per-

Sim	Scheme	ECB	EMEA	EPPS	GNOME	JRC	KDE4	NEWS	WIKI	avg
Base	-	49.23	49.53	42.83	49.99	59.05	49.52	**36.66**	35.15	46.50
FM	**tgt**^{1}	56.21	59.34	42.08	60.95	65.86	53.49	35.80	34.54	51.03
(Bulte and Tezcan, 2019)		585	765	195	686	612	575	54	184	457
FM	**tgt**^{1}**+STU**	**57.30**	61.03	42.95	62.68	67.24	54.68	35.54	35.16	52.07
(Xu et al., 2020)		585	765	195	686	612	575	54	184	457
FM	**s+t**^{1}	56.16	60.88	43.18	62.50	67.58	55.25	36.55	36.94	52.38
		585	765	195	686	612	575	54	184	457
S2V	**s+t**^{5}	57.16	60.44	**43.19**	62.44	65.39	51.32	35.98	35.82	51.47
		740	840	161	639	735	623	39	297	509
CBON	**s+t**^{5}	56.50	**61.04**	42.22	**63.76**	**68.75**	**55.83**	35.41	36.38	**52.49**
		710	896	195	854	733	862	63	378	586
FM	μ**adapt**	53.09	55.02	43.04	53.88	62.99	48.70	36.48	35.81	48.63
(Farajian et al., 2017)		585	765	195	686	612	575	54	184	457
CBON	μ**adapt**	53.41	53.32	43.20	54.77	63.37	52.06	36.47	36.39	49.12
(Farajian et al., 2017)		710	896	195	854	733	862	63	378	586

Table 3: BLEU scores for various model configurations and 8 test domains. Smaller numbers correspond to the number of input sentences in each domain for which at least one similar sentence is found.

sentence basis (Farajian et al., 2017; Li et al., 2018). We show micro-adaptation results when similar sentences are found by **CBON** and **FM** models (μ**adapt**). In our experiments, micro-adaptation does not yield the gains observed with priming methods. As previously stated, the best performing variants of the adaptation method presented in Farajian et al. (2017) were not included in our comparison. Variants employ a dynamically adapted learning rate and number of epochs.

Monolingual Corpora

Retrieval results shown in Table 3 (small font numbers) indicate a reduced number of similar sentences found for some domains (NEWS, EPPS and WIKI). In the context of scarce similar sentences, the boost in translation quality observed for most domains is subsequently reduced. The case of the **NEWS** domain is particularly harmful since worst results are always obtained when compared to our **Base** system.

However, very large monolingual collections of texts exist, far exceeding the amount of available parallel corpora. The latter are more expensive to collect and typically only exist for a limited number of domains and language pairs. With the objective to enhance NMT with monolingual corpora, we now apply the methods presented above to monolingual corpora.

We collect monolingual corpora in the target language (French in this work) and translate each sentence back into English to obtain synthetic parallel data. Similar to back-translation experiments in Sennrich et al. (2016b), we only use original (human-crafted) target-language data. We expect this to add less noise than incorporating synthetic target-language data into the NMT input. Once translated into English, the various priming approaches identify similar synthetic sentences and injects both the synthetic source and original target in the NMT input stream. Note that cross-lingual sentence embedding models exist (Sabet et al., 2019; Schwenk and Douze, 2017; Conneau and Lample, 2019) but our preliminary experiments using these tools did not show satisfactory results.

Thus, we exploit large collections of French texts for the News and Wikipedia domains (as detailed in Table 1) that we translate into English to enable similarity retrieval. Table 4 reports BLEU scores obtained by our best performing network **CBON** following the **s+t**^{5} scheme.

The supplementary number of similar sentences (468 input sentences have similar translations) collected for the WIKI domain over parallel and mono-

lingual[12] corpora (par+mon) yields an improvement of 2 BLEU points. However, very few (97) similar sentences are identified[13] over near 95 million sentences (par+mon), showing a small gain when compared to using only parallel sentences (par). The network does not succeed to outperform the accuracy of the **base** system. As outlined by Bulte and Tezcan (2019) and Xu et al. (2020) the accuracy of networks implementing priming may slightly drop in performance when no similar translations are integrated.

Sim	Scheme	Data	NEWS	WIKI
Base	-	-	**36.66**	35.15
CBON	**s+t**[5]	par	35.41 63	36.38 378
CBON	**s+t**[5]	par+mon	36.05 97	**38.20** 468

Table 4: Translation performance for the NEWS and WIKI domain test sets using similar sentences retrieved from parallel data (par) and from both parallel and monolingual (par+mon) data. The first two rows correspond to experiments already shown in Table 3.

5 Discussion

Unrelated Words

As previously outlined in Section 2, Xu et al. (2020) raised a problem regarding *unrelated* words. It concerns those words that, even through they appear in similar translations, must not be used to translate input sentences. An example of translation with unrelated word is given in Section 2.2 where the input sentence with similar translation:

vaccin contre la rougeole ∘ **pertussis vaccin**

is translated as: **vaccin contre la rougeole**, the right translation being: **vaccin contre la coqueluche**. The error is due to the fact that word *rougeole* is present in the input stream and is semantically related to *coqueluche*. The problem is particularly hurting when it involves keywords (like the proper noun in our example) which convey essential information regarding the meaning of sentences.

The work by Xu et al. (2020), that we denoted **tgt**[1]**+STU**, obtains an average reduction of these erroneous words in the translation hypotheses of 8%. We conduct the same experiment to analyse the performance of the new scheme **s+t**[1] introduced in this work. Table 5 reports the total number of unrelated words in 1-best similar sentences obtained by fuzzy matching[14]. As can be seen, the scheme **s+t**[1] further mitigates the apparition of unrelated words in translations, with a drop of -8.3%.

NMT Attention

We analyse the Encoder and Decoder self-attention layers, aiming to better understand how our **CBON s+t** model configuration makes use of similar translations.

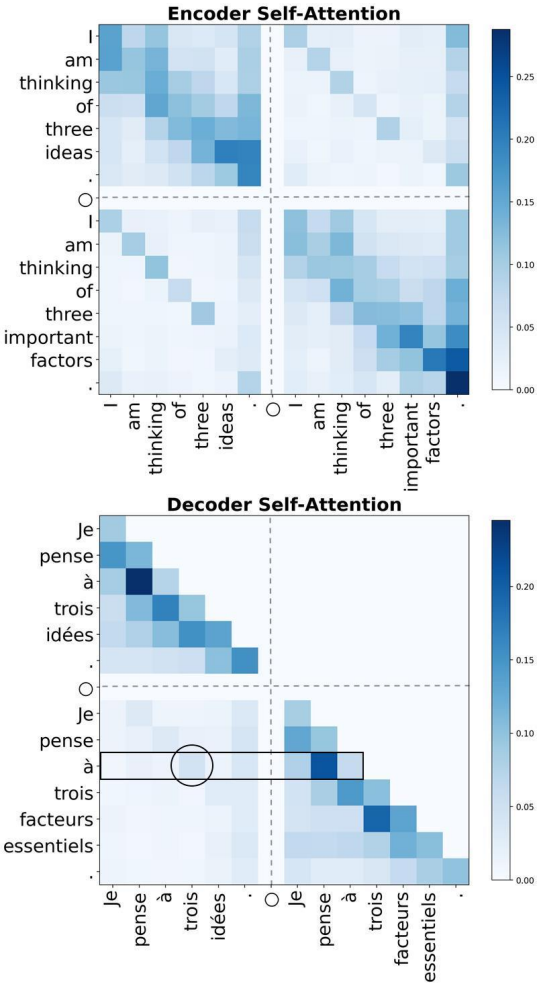


Figure 1: Average attention values of all heads through all layers for the encoder (top) and decoder (bottom). Dashed lines are used to separate similar and input sentences.

Figure 1 displays the attention[15] values for sen-

[12]Test French sentences entirely found in monolingual WIKI corpora are not considered as similar translations.

[13]In all cases we consider similar sentences s_i and s_j when $(\mathrm{EM}(s_i, s_j) \geq 0.8)$

[14]We follow the procedure detailed in Xu et al. (2020) to identify related/unrelated words.

[15]We use the average of all heads through all layers.

Scheme	ECB	EMEA	EPPS	GNOME	JRC	KDE4	NEWS	WIKI	avg
tgt^1+STU	3,555	2,320	312	1,285	3,515	940	39	344	1,538
s+t^1	3,199	1,985	306	1,195	3,413	845	31	310	1,410
unrelated	6,310	4,405	4,405	2,473	6,309	2,358	236	1,591	3,510

Table 5: Number of unrelated words appearing in test sets according to different augmentation schemes. The last row indicates the total number of unrelated words included in 1-best **FM** similar sentences.

tence s = [**I am thinking of three important factors .**] when translated into t = [**Je pense à trois facteurs essentiels .**] using the similar translation example s^1 = [*I am thinking of three ideas .*] and t^1 = [*Je pense à trois idées .*]. For visualization purposes we mask the attention of the sentence separator token ∘.

Concerning the encoder self-attention (top), we can clearly observe that the encoder pays attention to the words in the similar sentence (down-left) when embedding the input sentence (down-right). Equivalently, the decoder self-attention (bottom) also attends to the similar translation (down-left: prefix words generated in forced mode) when producing the translation of sentence s. Note that when the decoder is about to generate the French word *trois [three]*, attention weights (rectangle) are the highest for the preceding words (in particular to *pense [think]*), with *trois* (circle in the similar translation) also receiving a substantial weight. This suggests that the model has learned to use similar translations passed in the form of a target prefix to help generating translations.

Priming Model

The priming network leverages similar sentences from a TM so as to yield more accurate translations. From a mathematical perspective, the search for the best translation $\bar{t}$ is conditioned to the input sentence s as well as to similar pairs of translations s^1 and t^1:

$$\bar{t} = \arg\max_t P(t|s, s^1, t^1)$$

to facilitate reading we use one single similar translation (s^1 and t^1) rather than k-best translations.

To evaluate the intuition that $P(t|s, s^1, t^1)$ gives better translations than $P(t|s)$, we report the average of $\log P(t|s, s^1, t^1)$ computed by **CBON s+t^5** and of $\log P(t|s)$ computed by **Base** over test sets sentences with similar sentences translations.

Table 6 reports the difference between the token average of $\log P(t|s, s^1, t^1)$ and the token average of $\log P(t|s)$. More precisely, for each test sentence s, we compute the log probability of predicting reference t, we then sum all the calculated log probabilities and divide the sum by the total number of tokens in the references. For each test set, we computed the average log probability of model **CBON s+t^5** and **Base**. We report the difference in the average of both models. Results indicate that $\log P_{\textbf{CBON s+t}^5}(t|s^1, s, t^1)$ are actually greater than $\log P_{\textbf{Base}}(t|s)$ in most cases, with the exception of EPPS and NEWS for which the base system yields higher probabilities. We observe a strong correlation between values reported and the gap in BLEU score for the same model configurations.

Domain	**CBON s+t^5 − Base**
ECB	0.222
EMEA	0.231
EPPS	-0.039
GNOME	0.248
JRC	0.165
KDE4	0.252
NEWS	-0.173
WIKI	0.009

Table 6: Differences of token average log probability between **CBON s+t^5** and **Base** model.

Similarity over Synthetic Sentences

Results in Table 4 show a clear boost in performance (∼2 BLEU points) when making use of synthetic translations of the WIKI monolingual data set. We now want to measure the noise introduced by synthetic translations when compared to human translations. Thus, we consider the input sentences of the WIKI test set for which we found similar sentences in both the parallel (human translation) and monolingual (synthetic translation) corpus (279 sentences).

Results in Table 7 show a clear drop in BLEU scores when using synthetic matches. As expected, machine translation quality degrades the results

of similarity search which in turns provides less valuable similar translations.

Priming sentences	WIKI
par (human)	52.50
mon (synthetic)	49.94

Table 7: Results for a reduced test set (279 sentences) using **CBON** when priming with human and synthetic (back-translated) translations.

6 Related Work

Our work relates to the ideas introduced in Bulte and Tezcan (2019) and Xu et al. (2020). Both of them leverage similar translations from parallel corpora and inject similar sentences in the NMT network. While Bulte and Tezcan (2019) integrates fuzzy matches into the NMT model by concatenating similar translations to source sentences, Xu et al. (2020) extended the framework by adding additional source side features to distinguish between related and unrelated words, employed distributed sentence representations. A similar idea is also explored in Schwenk et al. (2019b), where the authors use multilingual sentence embeddings to retrieve pairs of similar sentences and train models uniquely with such sentences.

Previously, Niehues et al. (2016) augmented input sentences with pre-translations generated by a phrase-based MT system. Our work, in contrast, integrates similar sentences in both source and target sides and employs similar translations found in parallel as well as monolingual data sets.

A similar strategy of concatenating previous and current sentences was explored by Tiedemann and Scherrer (2017) further evaluated by Bawden et al. (2018) in the context of tackling discourse phenomena. Our work employs force decoding to allow including true translations in the decoder target-side. Thus, avoiding the error propagation problem (Ranzato et al., 2016) of longer sequences in auto-regressive models.

Bapna and Firat (2019) propose a neural MT model that incorporates retrieved neighbours relying on local phrase level similarities. Using deep pre-trained models (Peters et al., 2018; Radford et al., 2019; Devlin et al., 2019; Le et al., 2020; Conneau and Lample, 2019) to compute contextualized sentence representations has become common fashion in recent works (Feng et al., 2020; Chang et al., 2020). However, deep models suffer from computation complexity when applied on-the-fly for inference. We propose an extension of `sent2vec` (Pagliardini et al., 2018) to compute sentence representations that also inherits from the computationally efficient bilinear models (Mikolov et al., 2013a,b; Pennington et al., 2014).

Similar to our work, Farajian et al. (2017) and Li et al. (2018) retrieve similar sentence to dynamically adapt each individual input sentence. Farajian et al. (2017) obtains best performance when tuning the adaptation learning rate and number of epochs according to level of similarity between the input and retrieved sentences. In Xu et al. (2019) the model is dynamically adapted to a entire test set to reduce adaptation time.

In computer vision, priming network has been recently studied. For the object detection task , Rosenfeld et al. (2018) primed the network via an external information that affects all the processing layers. Upon processing each image in the network, Rosenfeld et al. (2018) also presented the network with the category of the object in the image; this information is injected at all layers.

7 Conclusions

Inspired by the human psychological phenomenon of priming, we have presented a simple framework for priming NMT networks. Following other research works, we used similar translations as priming cues to influence the NMT network. We presented a novel method that injects similar translations in the NMT network as prefixes of the decoder. The proposed method obtains higher translation accuracy results and reduces the undesirable effect observed in previous methods of copying unrelated words when performing translations.

We also proposed an extension to `sent2vec` that considers larger n-gram orders. It allows us to identify similar sentences (cues) that yield higher accuracy rates as measured on translation test sets.

We evaluate results on a multi-domain setting using a single model trained on a heterogeneous data set, built from multiple corpora and domains, achieving better results when compared to previous micro-adaptation approaches. In addition, we showed the suitability of our approach to gather valuable information from large monolingual corpora.

In our future work, we would like to explore alternative algorithms to compute distributed sentence representations from word embeddings, such

as TF-IDF. Furthermore, we would like to consider source sentence coverage when selecting n-best similar translations. As regards distributed representations we plan to experiment with cross-lingual networks to retrieve similar translations directly from human-crafted monolingual data in order to eliminate the noise introduced by synthetic translations.

Acknowledgments

This work was granted access to the HPC resources of [TGCC/CINES/IDRIS] under the allocation 2020- [AD011011270] made by GENCI (Grand Equipement National de Calcul Intensif). We also would like to thank the anonymous reviewers for their valuable suggestions.

References

Ankur Bapna and Orhan Firat. 2019. Non-parametric adaptation for neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1921–1931, Minneapolis, Minnesota. Association for Computational Linguistics.

Rachel Bawden, Rico Sennrich, Alexandra Birch, and Barry Haddow. 2018. Evaluating discourse phenomena in neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1304–1313, New Orleans, Louisiana. Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Bram Bulte and Arda Tezcan. 2019. Neural fuzzy repair: Integrating fuzzy matches into neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1800–1809, Florence, Italy. Association for Computational Linguistics.

Wei-Cheng Chang, Felix X. Yu, Yin-Wen Chang, Yiming Yang, and Sanjiv Kumar. 2020. Pre-training tasks for embedding-based large-scale retrieval.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 7059–7069. Curran Associates, Inc.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Georgiana Dinu, Prashant Mathur, Marcello Federico, and Yaser Al-Onaizan. 2019. Training neural machine translation to apply terminology constraints. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3063–3068, Florence, Italy. Association for Computational Linguistics.

M. Amin Farajian, Marco Turchi, Matteo Negri, and Marcello Federico. 2017. Multi-domain neural machine translation through unsupervised adaptation. In *Proceedings of the Second Conference on Machine Translation*, pages 127–137, Copenhagen, Denmark. Association for Computational Linguistics.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2020. Language-agnostic BERT sentence embedding.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Vancouver, Canada. Association for Computational Linguistics.

Rebecca Knowles and Philipp Koehn. 2016. Neural interactive translation prediction. In *Proceedings of the Conference of the Association for Machine Translation in the Americas (AMTA)*.

Catherine Kobus, Josep Crego, and Jean Senellart. 2017. Domain control for neural machine translation. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 372–378, Varna, Bulgaria. INCOMA Ltd.

Philipp Koehn and Jean Senellart. 2010. Convergence of Translation Memory and Statistical Machine Translation. In *Proceedings of AMTA Workshop on MT Research and the Translation Industry*, pages 21–31, Denver.

Hang Le, Loïc Vial, Jibril Frej, Vincent Segonne, Maximin Coavoux, Benjamin Lecouteux, Alexandre Allauzen, Benoît Crabbé, Laurent Besacier, and Didier Schwab. 2020. Flaubert: Unsupervised language model pre-training for french. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 2479–2490, Marseille, France. European Language Resources Association.

Xiaoqing Li, Jiajun Zhang, and Chengqing Zong. 2018. One sentence one model for neural machine translation. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*, Miyazaki, Japan. European Languages Resources Association (ELRA).

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

Jan Niehues, Eunah Cho, Thanh-Le Ha, and Alex Waibel. 2016. Pre-translation for neural machine translation. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1828–1836, Osaka, Japan. The COLING 2016 Organizing Committee.

Matteo Pagliardini, Prakhar Gupta, and Martin Jaggi. 2018. Unsupervised learning of sentence embeddings using compositional n-gram features. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 528–540, New Orleans, Louisiana. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2016. Sequence level training with recurrent neural networks. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Amir Rosenfeld, Mahdi Biparva, and John K. Tsotsos. 2018. Priming neural networks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2092–209209.

Ali Sabet, Prakhar Gupta, Jean-Baptiste Cordonnier, Robert West, and Martin Jaggi. 2019. Robust cross-lingual embeddings from parallel sentences.

Sebastin Santy, Sandipan Dandapat, Monojit Choudhury, and Kalika Bali. 2019. INMT: Interactive neural machine translation prediction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pages 103–108, Hong Kong, China. Association for Computational Linguistics.

Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán. 2019a. Wikimatrix: Mining 135m parallel sentences in 1620 language pairs from wikipedia. *CoRR*, abs/1907.05791.

Holger Schwenk and Matthijs Douze. 2017. Learning joint multilingual sentence representations with neural machine translation. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 157–167, Vancouver, Canada. Association for Computational Linguistics.

Holger Schwenk, Guillaume Wenzek, Sergey Edunov, Edouard Grave, and Armand Joulin. 2019b. Ccmatrix: Mining billions of high-quality parallel sentences on the web. *arXiv preprint arXiv:1911.04944*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Controlling politeness in neural machine translation via side constraints. In *Proceedings of*

the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 35–40, San Diego, California. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016c. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, pages 2214–2218, Istanbul, Turkey. European Languages Resources Association (ELRA).

Jörg Tiedemann and Yves Scherrer. 2017. Neural machine translation with extended context. In *Proceedings of the Third Workshop on Discourse in Machine Translation*, pages 82–92, Copenhagen, Denmark. Association for Computational Linguistics.

E Tulving, D.L. Schacter, and H.A. Stark. 1982. Priming effects in word-fragment completion are independent of recognition memory. *Journal of Experimental Psychology: Learning, Memory, & Cognition*, 8:336–342.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Jitao Xu, Josep Crego, and Jean Senellart. 2019. Lexical micro-adaptation for neural machine translation. In *International Workshop on Spoken Language Translation*, Honk Kong, China.

Jitao Xu, Josep Crego, and Jean Senellart. 2020. Boosting neural machine translation with similar translations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1580–1590, Online. Association for Computational Linguistics.

Subword Segmentation and a Single Bridge Language Affect Zero-Shot Neural Machine Translation

Annette Rios[1] and **Mathias Müller**[1] and **Rico Sennrich**[1,2]
[1]Department of Computational Linguistics, University of Zurich
[2]School of Informatics, University of Edinburgh

Abstract

Zero-shot neural machine translation is an attractive goal because of the high cost of obtaining data and building translation systems for new translation directions. However, previous papers have reported mixed success in zero-shot translation. It is hard to predict in which settings it will be effective, and what limits performance compared to a fully supervised system. In this paper, we investigate zero-shot performance of a multilingual EN↔{FR,CS,DE,FI} system trained on WMT data. We find that zero-shot performance is highly unstable and can vary by more than 6 BLEU between training runs, making it difficult to reliably track improvements. We observe a bias towards copying the source in zero-shot translation, and investigate how the choice of subword segmentation affects this bias. We find that language-specific subword segmentation results in less subword copying at training time, and leads to better zero-shot performance compared to jointly trained segmentation. A recent trend in multilingual models is to not train on parallel data between all language pairs, but have a single *bridge* language, e.g. English. We find that this negatively affects zero-shot translation and leads to a failure mode where the model ignores the language tag and instead produces English output in zero-shot directions. We show that this bias towards English can be effectively reduced with even a small amount of parallel data in some of the non-English pairs.

1 Introduction

Zero-shot translation has first been introduced by Firat et al. (2016) and refers to the ability of a multilingual NMT model to translate between all its source and target languages, even those pairs for which no parallel data was seen in training. In the simplest setting, all parameters in the network are shared between the different languages and the translation is guided only by special tags to indicate the desired output language (Johnson et al., 2017; Ha et al., 2016). While this capability is attractive because it is an alternative to building N^2 dedicated translation systems to serve N languages, performance on zero-shot pairs tends to lag behind pivot translation. Recent papers, such as Arivazhagan et al. (2019), Gu et al. (2019) and Zhang et al. (2020), have suggested training techniques to improve the generalization to unseen language pairs, but performance varies considerably across settings.

In this paper, we examine in detail the behavior of the multilingual model proposed by Johnson et al. (2017) on zero-shot translation directions. Our experiments show the following:

- Translation quality for zero-shot language pairs is highly unstable between different training runs, and between training checkpoints, which calls for more rigour to avoid false positive results.
- The incorrect copying of source text into the output is affected by the extent of subword copying at training time, and can be reduced by performing language-specific subword segmentation.
- English-centric models have a tendency to produce English text for non-English input. Multi-bridge models that include data from non-English pairs mitigate this problem.

Overall, we observe improvements of 8.1 BLEU (15.9→24.0) on 6 zero-shot directions with simple changes to the multilingual training setup.

2 Related Work

Our experiments are based on the multilingual model proposed by (Johnson et al., 2017; Ha et al.,

Proceedings of the 5th Conference on Machine Translation (WMT), pages 528–537
Online, November 19–20, 2020.

2016): A single model is trained on multiple language pairs with a standard encoder-decoder architecture, all parameters in the network are shared for all languages, including the vocabulary. An artificial target language token determines the output language. We prefix this special token to the source sentence as in Johnson et al. (2017). The major advantage of this model lies in its simplicity, since it does not require changing the architecture or training objective.

Several recent studies have explored approaches to improve generalization to zero-shot language pairs, for example through semi-supervised training (Gu et al., 2019; Currey and Heafield, 2019; Zhang et al., 2020) or alignment of encoder representations (Arivazhagan et al., 2019).

Our study is concerned with data conditions that enable zero-shot generalization for multilingual NMT, specifically preprocessing and data settings. While initial work used separate encoders and decoders for different languages (Firat et al., 2016), sharing of encoder and decoder parameters was established by Johnson et al. (2017); Ha et al. (2016) and has since been widely adopted. Johnson et al. (2017) use a shared subword segmentation model across languages, and this strategy is followed by later work (e.g. Aharoni et al., 2019; Zhang et al., 2020). Ha et al. (2016) do not share embeddings across languages, but use language-specific codes. We will show that both strategies cause errors.

In terms of data settings, the number of languages involved in multilingual models has increased from 3–4 (Firat et al., 2016; Johnson et al., 2017) to over 100 (Aharoni et al., 2019). The most popular setup are English-centric datasets, where the model is trained on translations between English and a number of other languages. A multi-way parallel corpus between 5 languages has been provided for the IWSLT17 multilingual task (Cettolo et al., 2012). Results on this dataset show strong zero-shot generalization, close or even exceeding the supervised condition (Lakew et al., 2017), but multi-way parallel corpora are only available in small amounts and specific domains, so we investigate alternatives to English-centric models that do not rely on multi-way parallelism.

3 Data and Models

Following Aharoni et al. (2019), our baseline setup is English-centric. For training, we use 5 million parallel sentences per language pair for English↔{French,Czech,German,Finnish} from WMT (Barrault et al., 2019). For all zero-shot language pairs, we sample test sets from OPUS (Tiedemann, 2012), see Table 1 for details.

To indicate the target language, we prefix a language tag on the source side (e.g. <2en>). Following Johnson et al. (2017), we segment all data with a byte-pair encoding model trained jointly on the training data in all five languages (Sennrich et al., 2016), with a threshold of 32k BPE operations. All our systems are base Transformers (Vaswani et al., 2017) implemented in Sockeye (Hieber et al., 2018), trained with early stopping based on BLEU on a development set that consists in equal parts of parallel sentences from all trained translation directions. See Appendix A and B for training details.

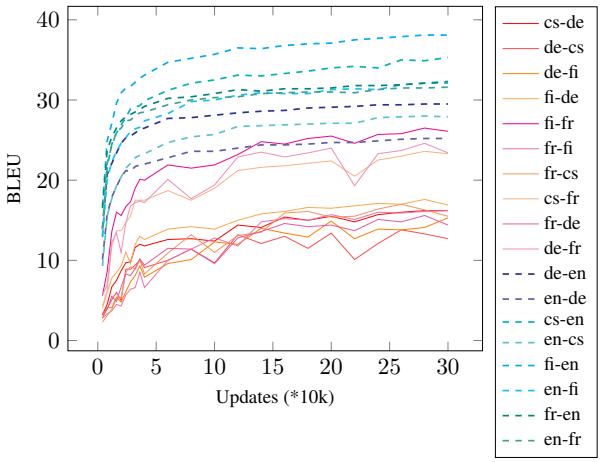


Figure 1: Baseline BLEU scores on test set as a function of training time. Dashed lines: trained pairs; solid lines: zero-shot pairs.

4 Baseline Experiments

BLEU[1] on zero-shot pairs is relatively unstable, see Fig. 1: while BLEU on the trained pairs increases steadily during training (dashed lines), performance on unseen language pairs fluctuates considerably, as also observed by Aharoni et al. (2019). Furthermore, multiple training runs result in relatively large differences in BLEU on the zero-shot directions. Across three training runs, average BLEU varies up to 0.24 points on trained language pairs (standard deviation: 0.12), but up to 6.28 BLEU on zero-shot pairs (standard deviation: 3.14) – see Table 2 for full results. We suspect that this fluctuation is due to the fact that the model is not op-

[1]SacreBLEU (Post, 2018): `BLEU+c.mixed+#.1 +s.exp+t.13a+v.1.2.21`.

	corpora	training	dev	test
Language Pairs with English:				
de↔en	Commoncrawl, Europarl-v9, Wikititles-v1	5M	250	2000
cs↔en	Europarl-v9, CzEng1.7	5M	250	2000
fr↔en	Commoncrawl, Europarl-v7	5M	250	2000
fi↔en	Europarl-v9, Wikititles-v1, Paracrawl-v3	4.35M*	250	2000
Multi-Bridge Pairs:				
fr↔fi	Rapid2016	350k	200	2000
cs↔de	Rapid2016, NewsCommentary, GlobalVoices	343k**	200	2000
Zero-shot test sets:				
de↔fi	Rapid2016			2000
de↔fr	Rapid2016, NewsCommentary, GlobalVoices			2000
cs↔fr	Rapid2016, NewsCommentary, GlobalVoices			2000

* oversampled to 5M
** oversampled to 350k

Table 1: Parallel corpora for training and testing. Sampled development sets are combined for training to a total of 2000 sentences (baselines) or 2800 sentences (training with cs↔de and fi↔fr). Europarl-v7, NewsCommentary and GlobalVoices retrieved from OPUS (Tiedemann, 2012), all other corpora are part of the WMT19 translation shared task (Barrault et al., 2019)

timized on zero-shot directions: models converge to different local minima that may be similarly good for trained pairs, but with no mechanism that stabilizes generalization to zero-shot pairs. If not stated otherwise, we will report the mean and standard deviation of three training runs with different seeds throughout the paper.

As an alternative to zero-shot translation, we report results obtained via pivot translation through English (e.g. German-English-Czech). On our data set, this approach works better than zero-shot translation. Pivot translation is stable across training runs, with a standard deviation of 0.19.

5 Copy Bias and Language-Specific Subword Segmentation

One failure mode we observe in zero-shot translation is over-copying of the input.[2] We suspect that for the translation of zero-shot directions, the model relies heavily on (sub-) words in the vocabulary that are shared between languages. To test this hypothesis, we train two models with language-specific subword segmentation:

a) a model with language-specific subword seg-

	Trained Directions		
	sampled test	official wmt test sets	
de-en	29.6 ±0.12	2019	31.9 ±0.35
cs-en	35.0 ±0.30	2018	25.7 ±0.20
fi-en	38.2 ±0.06	2019	25.2 ±0.20
fr-en	32.1 ±0.21	2015	33.9 ±0.56
en-de	25.2 ±0.21	2019	30.7 ±0.25
en-cs	28.4 ±0.45	2019	18.0 ±0.17
en-fi	32.1 ±0.20	2018	12.7 ±0.26
en-fr	31.7 ±0.21	2015	32.5 ±0.40
average	31.6 ±0.12		

	Zero-Shot Directions	
	direct	pivot
cs-de	14.7 ±1.39	20.3 ±0.32
de-cs	8.9 ±5.14	20.1 ±0.44
cs-fr	22.0 ±2.71	28.3 ±0.31
fr-cs	11.5 ±5.89	22.1 ±0.31
de-fr	23.3 ±2.48	29.0 ±0.15
fr-de	12.0 ±3.01	21.6 ±0.06
fi-fr	23.5 ±4.12	30.4 ±0.06
fr-fi	12.2 ±4.61	20.7 ±0.26
fi-de	15.1 ±1.69	21.3 ±0.15
de-fi	11.2 ±4.53	20.0 ±0.38
average	15.4 ±3.14	23.4 ±0.19

Table 2: Baseline (BLEU). Average and standard deviation of 3 training runs reported. For zero-shot directions, we compare direct zero-shot translation and pivot translation via English.

[2] See also (Ha et al., 2017; Arivazhagan et al., 2019; Zhang et al., 2020), who make similar observations in different settings.

mentation and no vocabulary overlap. We limit BPE operations to 10k per language.

b) similar to model a), with the exact same subword segmentation, but with vocabulary overlap.

For model a), we remove any potential vocabulary overlap by adding a language identifier to each subword. For instance, consider the preposition *in* in German and English: instead of one token *in*, the network vocabulary has an entry for *in#de#* and an additional entry for *in#en#*. This corresponds to the language-specific coding introduced by Ha et al. (2016).

For model b), we split words with the same language-specific BPE models as for a), but we allow vocabulary overlap, i.e. homographic forms in different languages are represented by a single entry in the network's vocabulary. This results in a vocabulary size of ∼50k for model a), whereas for model b), the vocabulary amounts to a total of ∼36k subwords.

Table 3 shows that removing vocabulary overlap does not affect the trained language pairs greatly, however, the effect on the zero-shot directions is quite harsh: For the first evaluation of model a), we remove only the correct target language tag (i.e. homographic forms with wrong language tag count as wrong), while for the second evaluation, we remove all language tags from the translations (i.e. homographic forms in other languages count as correct). In the first case, the model averages at only 4.7 BLEU on zero-shot directions, however, the more lenient second evaluation results in better scores (12.7 BLEU). This difference is due to the fact that the no-overlap model tends to produce a lot of English subwords (marked by #en#), especially for proper names and numbers.[3]

The second evaluation improves BLEU because the no-overlap model will often output the correct form, e.g. for proper names, if the word in the target language has the same spelling as in English.

Model b), with language-specific BPE and overlapping vocabularies, represents a compromise between a fully shared representation and fully language-specific coding. We hypothesize that allowing some vocabulary overlap helps aligning the representation between sentences with the same meaning in different languages, which is also supported by the effectiveness of cross-lingual pretraining with shared vocabularies for unsupervised MT and cross-lingual transfer (Conneau and Lample, 2019). We observe that models with jointly trained BPE develop a strong bias towards copying the input in zero-shot conditions. However, using language-specific BPE reduces the subword overlap between source and target sentences at training time, and consequently reduces this copying behavior at test time (see Table 4). Model b) indeed performs better (+5.1 BLEU) on the zero-shot directions than the original baseline with shared BPE (see Table 3).

[3]This essentially means that the strict evaluation gives us a more realistic estimate of the translation quality we can expect if the source and target language do not happen to share word forms, e.g. languages in different scripts.

	trained	zero-shot
jointly trained BPE	31.6 ±0.12	15.4 ±3.14
language-specific BPE:		
a) no overlap, strict*	30.9 ±0.58	4.7 ±1.90
a) no overlap, lenient**	31.3 ±0.59	12.7 ±2.52
b) vocabulary overlap	31.2 ±0.60	20.5 ±0.43

* homographic words in other languages=wrong
** homographic words in other languages=correct

Table 3: Average BLEU for models with language specific BPE, with and without vocabulary overlap.

	BPE	subwords	words
training set	jointly trained	9.70%	*5.70%
	lang.-specific	7.96%	*5.70%
translations	jointly trained	24.82%	20.58%
	lang.-specific	6.97%	4.70%

* identical

Table 4: Average word and subword overlap between source and target in training set, and in zero-shot translation output with jointly trained and language-specific BPE.

6 Multi-Bridge Models

A common issue in zero-shot translation is output in the wrong language. Previous work has addressed this with semi-supervised training (Gu et al., 2019; Arivazhagan et al., 2019; Zhang et al., 2020). We explore whether the recent trend to train English-centric models is to blame for this behavior. In most cases, the model will wrongly produce English output in zero-shot directions, since for all non-English languages, English was the only target language seen in training.

We suspect that adding even a small amount of parallel data in pairs without English will improve

	Single-Bridge	Multi-Bridge
	trained	
de-en	29.3 ±0.31	29.3 ±0.56
cs-en	35.1 ±0.81	34.9 ±0.65
fi-en	37.5 ±0.90	37.7 ±0.75
fr-en	31.5 ±0.42	31.6 ±0.30
en-de	24.9 ±0.38	24.9 ±0.40
en-cs	28.1 ±0.81	28.0 ±0.70
en-fi	31.6 ±0.60	31.6 ±0.70
en-fr	31.3 ±0.67	31.5 ±0.42
average	31.2 ±0.60	31.2 ±0.56
	zero-shot	trained
cs-de	17.6 ±0.30	21.7 ±0.60
de-cs	18.3 ±0.42	21.7 ±0.78
fi-fr	26.3 ±0.61	33.7 ±1.01
fr-fi	17.8 ±0.49	23.1 ±0.51
average	20.0 ±0.44	25.1 ±0.72
	zero-shot	
cs-fr	24.6 ±0.36	28.2 ±0.71
fr-cs	20.0 ±0.53	22.2 ±0.66
de-fr	26.6 ±0.30	29.5 ±0.31
fr-de	19.0 ±0.46	21.5 ±0.66
fi-de	18.3 ±0.35	21.6 ±0.60
de-fi	16.9 ±0.61	20.8 ±0.83
average	20.9 ±0.43	24.0 ±0.62

Table 5: BLEU for single-bridge baseline with language-specific BPE (see Table 3), and model trained with 350k pairs in de↔cs and fi↔fr (multi-bridge). Both models use language-specific BPE segmentation.

generalization, make models more sensitive to the language tag, and reduce the amount of English translations in the zero-shot directions. To test this hypothesis, we collect a small amount of parallel data in German-Czech and Finnish-French[4] and train our model with the additional language pairs. This new model has seen all non-English languages paired with exactly one other non-English language, but it still has zero-shot directions in de↔fr, fr↔cs and de↔fi. We use language-specific BPE segmentation and thus use the model with the best zero-shot performance from Table 3 as baseline.

The results in Table 5 show that even a small amount of parallel data in non-English language pairs increases generalization to unseen translation directions. The increase in BLEU scores for the newly added pairs de↔cs and fi↔fr are expected, but the new model also performs better on cs↔fr, de↔fr and fi↔de (+3.1 BLEU on average).

Following Zhang et al. (2020), we use the Python

[4]See Table 1 for details.

	single-bridge			multi-bridge		
	tgt	en	src	tgt	en	src
cs-fr	95.92	1.33	0.03	97.28	0.55	0
fr-cs	95.33	0.38	0.50	95.57	0.32	0.22
de-fr	94.00	1.72	1.40	95.43	0.97	0.82
fr-de	92.47	2.75	1.23	95.43	1.17	0.43
fi-de	91.65	2.40	0.60	94.38	0.88	0.33
de-fi	91.93	1.57	1.52	93.58	0.77	0.85
average	93.55	1.69	0.89	95.30	0.78	0.44

Table 6: Percentage of output produced in the correct target language (tgt), English, and the source language (src) in zero-shot translation according to automatic language identification. Models from Table 5.

version of langdetect[5] to estimate the number of translations in the correct language. Even though the amount of parallel data in de↔cs and fi↔fr was small compared to the directions with English (350k vs. 5 million sentence pairs), the new model is less likely to produce output in the wrong target language, as shown in Table 6.

7 Comparison to Back-Translation and Encoder Alignment

Previous work on the zero-shot generalization of multilingual NMT systems has proposed back-translation or changes to the training objective to improve translation in unsupervised directions. While we consider our proposed solutions on the data side to be complementary, and easier to adopt widely, we still want to discuss the question how our solutions compare to previous work, and whether they can be combined.

7.1 Back-Translation

Previous work has used fine-tuning with synthetic, back-translated data for translation directions that were unseen at training time (Gu et al., 2019; Currey and Heafield, 2019; Zhang et al., 2020). While this can mitigate the problem of producing output in the wrong language, this approach is sensitive to the zero-shot translation quality of back-translation.[6] We perform experiments following Gu et al. (2019) where we create synthetic corpora for all zero-resource directions via back-translations (250k sentences per translation direction), and fine-tune our models on the concatena-

[5]https://github.com/Mimino666/langdetect

[6]Unless back-translation is done via a pivot language, but note that Gu et al. (2019) report slightly better results for direct zero-shot back-translation.

	single-bridge	+align	+bt	multi-bridge	+align	+bt
en↔* (avg)	31.2 ±0.60	31.4 ±0.20	30.5 ±0.33	31.2 ±0.56	31.4 ±0.28	30.3 ±0.23
cs-fr	24.6 ±0.36	25.8 ±0.46	25.8 ±0.40	28.2 ±0.71	28.8 ±0.49	29.0 ±0.06
fr-cs	20.0 ±0.53	20.5 ±0.12	20.4 ±0.32	22.2 ±0.66	22.6 ±0.21	23.8 ±0.06
de-fr	26.6 ±0.30	27.4 ±0.26	26.1 ±0.38	29.5 ±0.31	29.7 ±0.47	30.8 ±0.17
fr-de	19.0 ±0.46	19.6 ±0.15	19.8 ±0.21	21.5 ±0.66	21.6 ±0.31	22.0 ±0.00
de-fi	16.9 ±0.61	17.9 ±0.26	17.4 ±0.29	20.8 ±0.83	21.2 ±0.50	21.9 ±0.29
fi-de[1]	18.3 ±0.35	18.8 ±0.12	*20.1 ±0.40*	21.6 ±0.60	22.0 ±0.23	*23.9 ±0.00*
average	20.9 ±0.43	21.7 ±0.19	21.6 ±0.30	24.0 ±0.62	24.3 ±0.36	25.2 ±0.03

[1] used as development set for early stopping for +bt

Table 7: Zero-resource translation performance (BLEU) with single-bridge and multi-bridge multilingual models, fine-tuned with a cosine loss to reward encoder representation alignment (+align), and back-translation for zero-resource translation directions (+bt).

tion of this data, plus 250k sentence pairs per supervised translation direction. As base system for both back-translation and fine-tuning, we consider both our single-bridge and our multi-bridge system.

As stopping criterion during fine-tuning, we use BLEU on the Finnish↔German test set, one of the zero-resource language pairs. This leaves us with 5 translation directions that are still purely zero-resource.

7.2 Encoder Alignment

Arivazhagan et al. (2019) propose to use cosine distance as an additional loss term for multilingual models. The cosine distance loss encourages the model to produce encoder representations for sentences in the source language that are similar to the encoder representation of the same sentence in the target language. This, directly and indirectly, rewards similarity of encoder representations across all languages. We implement cosine loss in Sockeye, but instead of normalising sequence lengths by max pooling like Arivazhagan et al. (2019), we average encoder states, as proposed by Gouws et al. (2015). We introduce a new hyperparameter λ that scales cosine distance (CD) loss w.r.t. the standard cross-entropy (CE):

$$\mathcal{L} = (1 - \lambda) * CE + \lambda * CD \quad (1)$$

We train models with $\lambda = 0.5$. As in our experiments with back-translation, we do not train from scratch, but fine-tune each of the single-bridge and multi-bridge models with a patience of 10.[7]

7.3 Results

Results are shown in Table 7. The gains from using more than one bridge language and back-translation are cumulative: Both the single- and the multi-bridge baseline improve with encoder alignment and back-translation, but the multi-bridge performs better overall in zero-resource directions.

Aligning encoder representations leads to an increase of 0.8 BLEU for the zero-shot directions for the single bridge data. In the multi-bridge scenario however, the effect of the additional loss is smaller (+0.3 BLEU on average). Table 7 contains only results for models with language-specific subword segmentation; but preliminary experiments show that aligning encoder representations of one of the baselines from Table 2 with jointly trained BPE gives a similar result: Encoder alignment alone does not fix the underlying issue caused by vocabulary overlap and English-centric models, even though we observe an increase of $\sim$ 1.5 BLEU points in zero-shot directions over the baseline.

Back-translation leads to an average improvement of 0.7 BLEU with single-bridge data, and 1.2 BLEU with multi-bridge data. On the fully supervised pairs English↔{Czech,German,Finnish,French}, we observe a performance drop by 0.7–0.9 BLEU with back-translation. Again, back-translation alone does not seem to solve the issues of single-bridge setups, and the model benefits from additional supervised translation directions.

On the 6 remaining zero-shot translation direc-

[7] In a new training run with random initialization, the encoder produces highly similar representations for all languages from the start. Arivazhagan et al. (2019) report that fine-tuning yields better results.

tions, our pivoting baseline (Table 2) achieves an average BLEU of 23.7. Our best system with multi-bridge data and back-translation achieves 25.2, and thus outperforms our pivoting baseline by 1.5 BLEU.

8 Conclusions

We analyze the importance of shared subwords in multilingual models and find that language-specific BPE segmentation helps to reduce the amount of untranslated segments in zero-shot directions. Furthermore, we explore whether the tendency to produce the wrong output language can be attributed to using English as the only bridge language, and show that even with a small amount of additional training data in non-English language pairs, generalization to unseen translation directions improves as the model is less likely to produce output in the wrong language.

Compared to previous work, the methods we propose are easier to implement, since they only concern data collection and pre-processing, and result in higher gains for zero-shot directions. They are also compatible in principle with approaches that introduce new training objectives or model modifications, and we report best results when fine-tuning a multi-bridge model with back-translation for zero-resource translation directions.

For future work, we are interested in testing the effects of subword regularization (Kudo, 2018; Provilkov et al., 2020) on zero-shot translation performance, and scaling multi-bridge setups to massively multilingual settings.

Acknowledgements

This work has received funding from the Swiss National Science Foundation (SNF, grants 105212_169888 and 176727).

References

Roee Aharoni, Melvin Johnson, and Orhan Firat. 2019. Massively Multilingual Neural Machine Translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3874–3884, Minneapolis, Minnesota. Association for Computational Linguistics.

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Roee Aharoni, Melvin Johnson, and Wolfgang Macherey. 2019. The Missing Ingredient in Zero-Shot Neural Machine Translation. *CoRR*, abs/1903.07091.

Loïc Barrault, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. 2019. Findings of the 2019 Conference on Machine Translation (WMT19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 1–61, Florence, Italy. Association for Computational Linguistics.

Mauro Cettolo, Christian Girardi, and Marcello Federico. 2012. Wit^3: Web inventory of transcribed and translated talks. In *Proceedings of the 16th Conference of the European Association for Machine Translation (EAMT)*, pages 261–268, Trento, Italy.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual Language Model Pretraining. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 7059–7069. Curran Associates, Inc.

Anna Currey and Kenneth Heafield. 2019. Zero-resource neural machine translation with monolingual pivot data. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 99–107, Hong Kong. Association for Computational Linguistics.

Orhan Firat, Baskaran Sankaran, Yaser Al-onaizan, Fatos T. Yarman Vural, and Kyunghyun Cho. 2016. Zero-Resource Translation with Multi-Lingual Neural Machine Translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 268–277, Austin, Texas. Association for Computational Linguistics.

Stephan Gouws, Yoshua Bengio, and Greg Corrado. 2015. Bilbowa: Fast bilingual distributed representations without word alignments. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 748–756, Lille, France. PMLR.

Jiatao Gu, Yong Wang, Kyunghyun Cho, and Victor O.K. Li. 2019. Improved Zero-shot Neural Machine Translation via Ignoring Spurious Correlations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1258–1268, Florence, Italy. Association for Computational Linguistics.

Thanh-Le Ha, Jan Niehues, and Alexander H. Waibel. 2016. Toward Multilingual Neural Machine Translation with Universal Encoder and Decoder. In *International Workshop on Spoken Language Translation 2016*, Seattle, USA.

Thanh-Le Ha, Jan Niehues, and Alexander H. Waibel. 2017. Effective strategies in zero-shot neural machine translation. In *International Workshop on Spoken Language Translation 2017*, Tokyo, Japan.

Felix Hieber, Tobias Domhan, Michael Denkowski, David Vilar, Artem Sokolov, Ann Clifton, and Matt Post. 2018. The Sockeye Neural Machine Translation Toolkit at AMTA 2018. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (AMTA)*, volume 1: Research Papers, pages 200–207.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Taku Kudo. 2018. Subword Regularization: Improving Neural Network Translation Models with Multiple Subword Candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia. Association for Computational Linguistics.

Surafel Melaku Lakew, Quintino Lotito, Marco Turchi, Matteo Negri, and Marcello Federico. 2017. FBK's Multilingual Neural Machine Translation System for IWSLT 2017. In *International Workshop on Spoken Language Translation 2017*, pages 113–119, Tokyo, Japan.

Matt Post. 2018. A Call for Clarity in Reporting BLEU Scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

Ivan Provilkov, Dmitrii Emelianenko, and Elena Voita. 2020. BPE-Dropout: Simple and Effective Subword Regularization. In *Proceedings of the 2020 Annual Conference of the Association for Computational Linguistics*. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725. Association for Computational Linguistics.

Jörg Tiedemann. 2012. Parallel Data, Tools and Interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, pages 2214–2218, Istanbul, Turkey. European Languages Resources Association (ELRA).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 6000–6010. Curran Associates, Inc.

Biao Zhang, Philip Williams, Ivan Titov, and Rico Sennrich. 2020. Improving Massively Multilingual Neural Machine Translation and Zero-Shot Translation. In *Proceedings of the 2020 Annual Conference of the Association for Computational Linguistics*. Association for Computational Linguistics.

A Model Size and Training

All models are trained with the Sockeye toolkit (Hieber et al., 2018)[8] on 5 Tesla-V100 (16GB) GPUs for 4-5 days.

model type	number of parameters
1 joint bpe baseline	60,516,602
2 language specific bpe, no vocabulary overlap	69,728,543
3 language specific bpe, vocabulary overlap	62,470,619
4 language specific bpe, vocabulary overlap, multi-bridge	62,490,626

Table 8: Number of Parameters per Model Type. Numbers vary between models due to different vocabulary sizes. Vocabulary is built automatically based on training data, therefore, 4 has a slightly larger vocabulary than 3. Cosine-loss models have the same number of parameters as 3 and 4.

model type	best checkpoint and BLEU					
	seed=1		seed=2		seed=3	
1 joint bpe baseline	106	30.7	95	30.9	94	31.3
2 language specific bpe, no vocabulary overlap	66	29.3	115	31.0	60	30.0
3 language specific bpe, vocabulary overlap	90	30.3	120	31.0	55	29.4
4 language specific bpe, vocabulary overlap, multi-bridge	117	29.3	80	28.6	59	28.2

Table 9: Best checkpoint according to BLEU on development set (patience=10). Sentence pairs in the development sets are identical for each model, however the dev set for model 4 contains additional samples in cs↔de and fi↔fr.

[8] https://github.com/awslabs/sockeye

B Hyperparameters

Training Hyperparameters for all Models	
training settings:	
batch type	word
batch size	16384
max-seq-len	100:100
word-min-count	1:1
seed	1/2/3
model settings:	
encoder	transformer
decoder	transformer
num-layers	6:6
transformer-model-size	512
transformer-attention-heads	8
transformer-feed-forward-num-hidden	2048
transformer-preprocess	n
transformer-postprocess	dr
transformer-positional-embedding-type	fixed
num-embed	512:512
weight-tying-type	src_trg_softmax
optimization settings:	
optimizer	adam
optimized-metric	bleu
checkpoint interval	4000
max-num-checkpoint-not-improved	10
min-num-epochs	0
max-updates	1001000
label-smoothing	0.1
gradient-clipping-threshold	-1
initial-learning-rate	0.0001
learning-rate-reduce-num-not-improved	8
learning-rate-reduce-factor	0.7
learning-rate-scheduler-type	plateau-reduce
learning-rate-warmup	0
initialization settings:	
weight-init	xavier
weight-init-scale	3.0
weight-init-xavier-factor-type	avg
dropout settings:	
embed-dropout	0:0
transformer-dropout-attention	0.1
transformer-dropout-act	0.1
transformer-dropout-prepost	0.1

Table 10: Sockeye hyperparameters for all models (values with ':' = encoder:decoder)

Look It Up: Bilingual and Monolingual Dictionaries Improve Neural Machine Translation

Xing Jie Zhong
University of Notre Dame
xzhong3@nd.edu

David Chiang
University of Notre Dame
dchiang@nd.edu

Abstract

Despite advances in neural machine translation (NMT) quality, rare words continue to be problematic. For humans, the solution to the rare-word problem has long been dictionaries, but dictionaries cannot be straightforwardly incorporated into NMT. In this paper, we describe a new method for "attaching" dictionary definitions to rare words so that the network can learn the best way to use them. We demonstrate improvements of up to 3.1 BLEU using bilingual dictionaries and up to 0.7 BLEU using monolingual source-language dictionaries.

1 Introduction

Despite its successes, neural machine translation (NMT) still has unresolved problems. Among them is the problem of rare words, which are paradoxically very common because of Zipf's Law. In part, this is a problem intrinsic to data-driven machine translation because the system will inevitably encounter words not seen in the training data. In part, however, NMT systems seem particularly challenged by rare words, compared with older statistical models.

One reason is that NMT systems have a fixed-size vocabulary, typically 10k–100k words; words outside this vocabulary are represented using a special symbol like UNK. Byte pair encoding (BPE) breaks rare words into smaller, more frequent subwords, at least allowing NMT to see them instead of UNK (Sennrich et al., 2016). But this by no means solves the problem; even with subwords, NMT seems to have difficulty learning translations of very rare words, possibly an instance of catastrophic forgetting (McCloskey and Cohen, 1989).

Humans deal with rare words by looking them up in a dictionary, and the idea of using dictionaries to assist machine translation is extremely old. From a statistical perspective, dictionaries are a useful complement to running text because the uniform distribution of dictionary headwords can smooth out the long-tailed distribution of running text. In pre-neural statistical machine translation systems, the typical way to incorporate bilingual dictionaries is simply to include them as parallel sentences in the training data. But (as we show), this does not work well for NMT systems.

We are aware of only a few previous attempts to find better ways to incorporate bilingual dictionaries in NMT. Some methods use dictionaries to synthesize new training examples (Zhang and Zong, 2016; Qi et al., 2018; Hämäläinen and Alnajjar, 2019). Arthur et al. (2016) extend the model to encourage it to generate translations from the (automatically extracted) dictionary. Post and Vilar (2018) constrain the decoder to generate translations from the dictionary. What these approaches have in common is that they all treat dictionary definitions as target-language text, when, in fact, they often have properties very different from ordinary text. For example, CEDICT defines 此致 (*cǐzhì*) as "(used at the end of a letter to introduce a polite salutation)" which cannot be used as a translation. In the case of a monolingual source-language dictionary, the definitions are, of course, not written in the target language at all.

In this paper, we present an extension of the Transformer (Vaswani et al., 2017) that "attaches" the dictionary definitions of rare words to their occurrences in source sentences. We introduce new position encodings to represent the nonlinear structure of a source sentence with its attachments. Then the unmodified translation model can learn how to make use of this attached information. We show that this additional information yields improvements in translation accuracy of up to 3.1 BLEU. Because our method does not force dictionary definitions to be treated as target-language text, it is generalizable to other kinds of information, such as monolingual source-language dictionaries, which yield smaller improvements, but still as much as 0.7 BLEU.

Proceedings of the 5th Conference on Machine Translation (WMT), pages 538–549
Online, November 19–20, 2020.

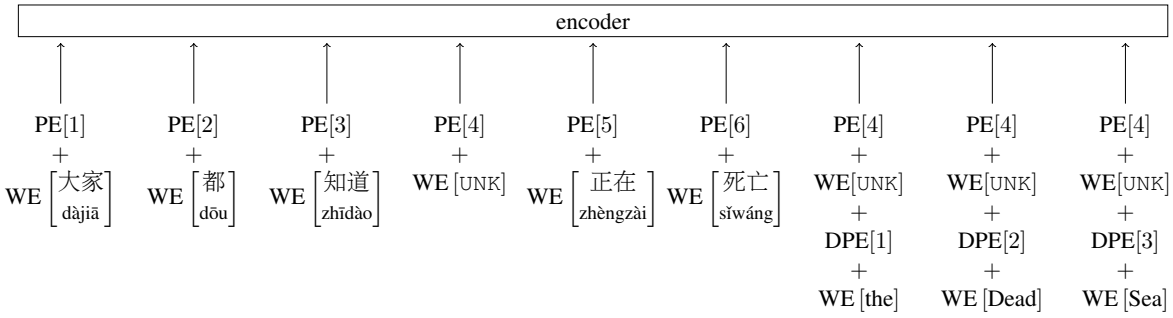


Figure 1: Our method attaches dictionary definitions to rare words. Here, the source sentence is 大家 都 知道 死海 正在 死亡 (*dàjiā dōu zhīdào Sǐhǎi zhèngzài sǐwáng*, *Everyone knows that the Dead Sea is dying*). WE$[f]$ is the embedding of word f, PE$[p]$ is the encoding of position p, and DPE$[q]$ is the encoding of position q within a dictionary definition. The rare word 死海 (*Sǐhǎi*) is replaced with UNK and defined as *the Dead Sea*. The words of the definition are encoded with both the position of the defined word (4) and their positions within the definition.

2 Methods

Our method is built on top of the Transformer (Vaswani et al., 2017). For each unknown source word with an entry in the dictionary, we attach the first 50 tokens of the definition (discarding the rest of the definition) to the source sentence. As described below, we encode the definition so as to differentiate it from the source sentence proper and to record which source word the definition is attached to. We leave the task of deciding whether and how to use the definition up to the translation model, which we use without any modifications.

2.1 Position encodings

To differentiate the attached definitions from the source sentence itself, we use special position encodings.

An ordinary word f at position p is encoded, as usual, as $\mathrm{E}[f] = \mathrm{WE}[f] + \mathrm{PE}[p]$, where WE is the word embedding and PE is the usual sinusoidal position encoding (Vaswani et al., 2017).

Suppose that word f at position p has an attached definition. Then word d at position q of the definition is encoded as

$$\mathrm{E}[d] = \mathrm{WE}[f] + \mathrm{PE}[p] + \mathrm{WE}[d] + \mathrm{DPE}[q],$$

where DPE is a position encoding scheme different from PE. We experimented with several schemes for DPE; in the experiments below, we learned a different encoding for each position (Gehring et al., 2017).

See Figure 1 for an illustration of the encoding of an example source sentence. Note that once all words have received their position encodings, their order does not matter, as the Transformer encoder is order-independent.

2.2 Subword segmentation

To apply our method on data that has been segmented using BPE, we face two new problems. First, since very few words are replaced with UNK, it is not sufficient only to attach definitions to UNK. How do we decide which words to attach definitions to? Second, if a word has been split into multiple subwords, the definition does not have a single attachment position. How do we represent the attachment position when encoding the definition?

To choose which words to define, we use a simple frequency threshold. We scan the data (after tokenization/segmentation but before BPE) for matches with the dictionary, including multi-word matches. If any substring of the source sentence matches a headword in the dictionary and occurs in the training data k or fewer times, we attach its definition. The threshold k can be tuned on the development data.

To attach a definition to a substring with more than one token, we simply fuse all the tokens in the substring into a single token, which often (but not always) then falls out of the vocabulary and is therefore changed to UNK. We attach the dictionary definition to this single token, which represents the whole word or expression.

For example, in the sentence in Figure 1, BPE splits 死海 (*sǐhǎi*) into 死@@ 海 (*sǐ@@ hǎi*) (where @@ is the marker that typical implementations of BPE use to indicate subword splits). Assuming that 死海 occurs k or fewer times, we fuse it back into a single token, which gets changed into UNK. Then the dictionary definition is attached as described above.

Language	Task	lines train	dev	test	total	words tokens	types	vocab
Chi-Eng	Spoken	176,000	22,000	22,000	220k	5.9M	179k	25k
	Science	216,000	27,000	27,000	270k	10.1M	383k	27k
	Laws	176,000	22,000	22,000	220k	17.4M	98k	22k
	News	360,000	45,000	45,000	450k	25.3M	477k	24k
	Education	360,000	45,000	45,000	450k	18.6M	461k	28k
	Subtitles	240,000	30,000	30,000	300k	6.6M	147k	27k
	Thesis	240,000	30,000	30,000	300k	17.2M	613k	27k
	UM-all	1,993,500	221,500	5,000	2.2M	101.3M	1.3M	33k
Deu-Eng	Europarl-small	160,000	20,000	20,000	200k	10.9M	151k	16k
	Europarl-all	1,440,000	180,000	197,758	1.8M	98.6M	475k	16k

Table 1: Statistics of the various tasks we experimented on. Train/dev/test: number of lines selected for use as training, development, and test data (respectively). Toks: number of word tokens (source+target). Types: number of word types (source+target). Vocab: joint vocabulary size used in word-based experiments.

3 Experiments

In this section, we describe our experiments on Chinese-English and German-English translation, comparing our method (which we call *Attach*) against two baselines. One baseline is the standard Transformer without any dictionary information (which we call *Baseline*). The other baseline is the standard Transformer with the bilingual dictionaries included as parallel sentences in the training data (which we call *Append*).

3.1 Data: Chinese-English

For Chinese-English, we used the UM-Corpus[1] (Tian et al., 2014), which has about 2M sentence pairs in eight different domains. Since rare words may be more frequent in certain domains, testing our model on different types of data may highlight the conditions where dictionaries can be helpful. We excluded the Microblog domain because of its length (only 5000 lines). For each of the other domains, we split the data into three parts: the first roughly 80% for training (*train*), the next 10% for development (*dev*), and the last 10% for testing (*test*). The task *UM-all* combines all eight domains. The UM-Corpus provides a test set, which we used (*test*), and we split the provided training data into two parts, the first 90% for training (*train*) and last 10% for development (*dev*). The exact line counts and other statistics are shown in Table 1.

We used the Stanford segmenter[2] (Chang et al., 2008) for the Chinese data and the Moses tokenizer[3] for the English data.

As a dictionary, we used CC-CEDICT,[4] which has 116,493 entries. Each entry has a traditional Chinese headword (which we delete), a simplified Chinese headword, a pronunciation (which we delete), and one or more definitions. We process the definitions as follows:

- Remove substrings of the form *abbr. for* c, where c is a Chinese word.
- If a definition contains *see* c or *see also* c, where c is a Chinese word, replace it with the definition of c.
- Remove everything in parentheses.
- Remove duplicate definitions.
- If the entry has no definitions left, delete the whole entry.
- Concatenate all the definitions into a single string.

The resulting dictionary has 102,567 entries, each consisting of a Chinese headword and a single English definition. We segmented/tokenized these in the same way as the parallel data. The average definition length is five, and the maximum definition length is 107.

[1] http://nlp2ct.cis.umac.mo/um-corpus/

[2] https://nlp.stanford.edu/software/segmenter.shtml

[3] http://www.statmt.org/moses/

[4] https://www.mdbg.net/chinese/dictionary?page=cedict, downloaded 10/2018.

For example, consider the following CEDICT entries, where we have already removed traditional Chinese characters and pronunciations for clarity.

三自 /abbr. for 三自爱国教会, Three-Self Patriotic Movement/
U盘 /USB flash drive/see also 闪存盘
闪存盘 /USB flash drive/jump drive/thumb drive/memory stick/

After cleaning, these would become

三自 Three-Self Patriotic Movement
U盘 USB flash drive jump drive thumb drive memory stick
闪存盘 USB flash drive jump drive thumb drive memory stick

3.2 Data: German-English

For German-English, we used the Europarl V7 dataset.[5] We tokenized both sides of the data with the Moses tokenizer. Due to the size of the original Europarl dataset and the increased runtime from our method, we ran some experiments on only the first 200k lines of the dataset, denoted in result tables as *Europarl-small*, while the full Europarl data is called *Europarl-all*. We split both into three parts: the first roughly 80% for training, the next 10% for development, and the last 10% for testing. Some statistics of the data are shown in Table 1.

We used the German-English dictionary from Stardict,[6] which is derived from Freedict[7] and has 81,628 entries. In this dictionary, the headwords have notes in parentheses indicating things like selectional restrictions; we deleted all of these. Unlike with CEDICT, we did not delete any material in definitions, nor did we resolve cross-references, which were very rare. As before, we removed blank entries and merged multiple definitions into a single line. We tokenized both headwords and definitions with the Moses tokenizer. The final dictionary size is 80,737 entries, with an average definition length of 2.9 and a maximum definition length of 88.

For example, the entry:

(Aktien) zusammenlegen to merge (with)

would become

zusammenlegen to merge (with)

[5] http://statmt.org/europarl/
[6] http://download.huzheng.org/freedict.de/
[7] https://freedict.org/

Task	Baseline	Append	Attach
Spoken	13.6	12.4	15.4
Science	8.0	6.6	9.2
Laws	29.0	27.4	30.2
News	9.9	10.2	11.2
Education	9.1	8.7	9.9
Subtitles	18.3	16.4	20.2
Thesis	9.5	9.5	10.6
UM-all	16.8	16.7	17.7
Europarl-small	29.2	28.4	29.6
Europarl-all	30.0	29.8	30.1

Table 2: Results on word-based translation. Our method (Attach) significantly improves over the baseline in all tasks. Appending the dictionary to the parallel data (Append) performs worse in all tasks except in News; differences are significant for all tasks except UM-all and Thesis.

3.3 Implementation and details

We used Witwicky,[8] an open-source implementation of the Transformer, with all of its default hyperparameters. We use the same random seed in each experiment. We modified it to attach dictionary definitions as described above. The code and our cleaned dictionaries are available under an open-source license.[9]

For BPE-based translation, we used joint BPE with 16k operations. For word-based translation, we set each system's vocabulary size close to the vocabulary size of the corresponding BPE-based system. For example, the Spoken dataset with 16k BPE applied to the training data has 25,168 word types, so we limited the word-based model to 25,000 word types. The vocabulary size we chose for each data set is shown in Table 1.

For all tasks except UM-all and Europarl-all, we trained for 20 epochs, and used the model with the highest dev BLEU to translate the test set. Due to the massive increase in training data on the UM-all and Europarl-all datasets, we only trained for 10 epochs. Otherwise, the settings are the same across all experiments.

We report case-insensitive BLEU scores of detokenized outputs against raw references. We perform significance testing with bootstrap resampling using 1000 samples, with a significance level of 0.05.

[8] https://github.com/tnq177/witwicky
[9] https://github.com/xjz92/Attach-Dictionary

Method	UM-Spoken Dev BLEU
Baseline	13.6
Attach to unknown words	13.9
+ fused multi-word expressions	13.8
+ all words	13.8

Table 3: Comparison of variations of our method on word-based translation.

Method	UM-Spoken Dev BLEU
Baseline	14.2
Attach to fused unknown words	14.8
+ fused multi-word expressions	14.8

Table 4: Comparison of variations of our method on BPE-based translation.

3.4 Results: Word-Based

Table 2 shows results on word-based translation. The *Append* column shows that simply appending the bilingual dictionary to the parallel training data is not helpful, for all tasks except News; these differences are significant for all tasks except UM-all and Thesis. By contrast, our method improves accuracy significantly over the baseline across all tasks.

We also compared against some variations of our method. First, CEDICT has definitions for single words as well as multi-word expressions. In our original setup, we only look up unknown single words, so the definitions for multi-word expressions are never used. To fully utilize the dictionary, we tried changing the source data by taking every substring that matches a dictionary entry and fusing it into a single token, which would often, but not always, fall out of the vocabulary and be changed to `UNK`. When more than one match was possible, we chose the longest possible match, breaking ties arbitrarily. However, we found that fusing phrases did not perform as well as just fusing words (Table 3). We also tried attaching dictionary definitions to all tokens, not just `UNK` tokens. Unfortunately, this also did not perform as well (Table 3).

3.5 Results: BPE-Based

As described in Section 2.2, we fuse subwords in order to attach definitions. Again we must first decide whether we wanted to fuse multi-word expressions.

Task	Baseline	Append	Fuse	Attach
Spoken	16.6	14.7	16.3	17.0
Science	11.6	9.6	13.8	14.7
Laws	29.0	26.8	29.0	30.0
News	11.8	10.9	11.3	13.3
Education	12.9	12.3	12.2	14.2
Subtitles	20.0	17.3	19.7	21.3
Thesis	15.3	14.2	14.9	15.5
UM-all	19.8	19.7	19.3	21.4
Europarl-small	32.6	30.8	33.4	33.5
Europarl-all	35.3	36.0	36.1	36.5

Table 5: Results on BPE-based translation. Our method (Attach) improves significantly over the baseline in Europarl-small and all Chinese-English tasks, whereas appending the dictionary to the parallel data (Append) performs worse, significantly so for Europarl-small and all Chinese-English tasks except UM-all. For Europarl-all, Append is significantly better. The Fuse column shows the effect of fusing words that would receive definitions, without actually attaching the definitions.

On the dev set, both methods have comparable performance (Table 4). Since we were interested in using as much of the dictionary as possible, we chose the model that fuses phrases.

As described in Section 2.2, we fuse subwords and attach definitions only for words whose frequency falls below a threshold. To tune this threshold, we trained models using thresholds of $k = 5$, 10, 15, 20, 50, 100, and ∞, and measured BLEU on the development set (Figure 2). We found that for Chinese-English, $k = \infty$ was best, but for German-English, $k = 5$ was best.

The results are shown in Table 5. As before, we compared against the two baselines (*Baseline* and *Append*). To tease apart the effect of fusing words and adding dictionary definitions, we also tested a model where all words that would receive definitions are fused, but the definitions are not actually attached (*Fuse*). Finally, we tested our model (*Attach*). On Chinese-English, our model improved significantly over the baselines across all tasks, whereas appending the dictionary to the parallel data did worse, significantly so on all tasks except UM-all. On German-English, the results on Europarl-small were similar, with Append doing significantly worse and our model doing significantly better. Interestingly, on Europarl-all, Append does significantly better than the baseline.

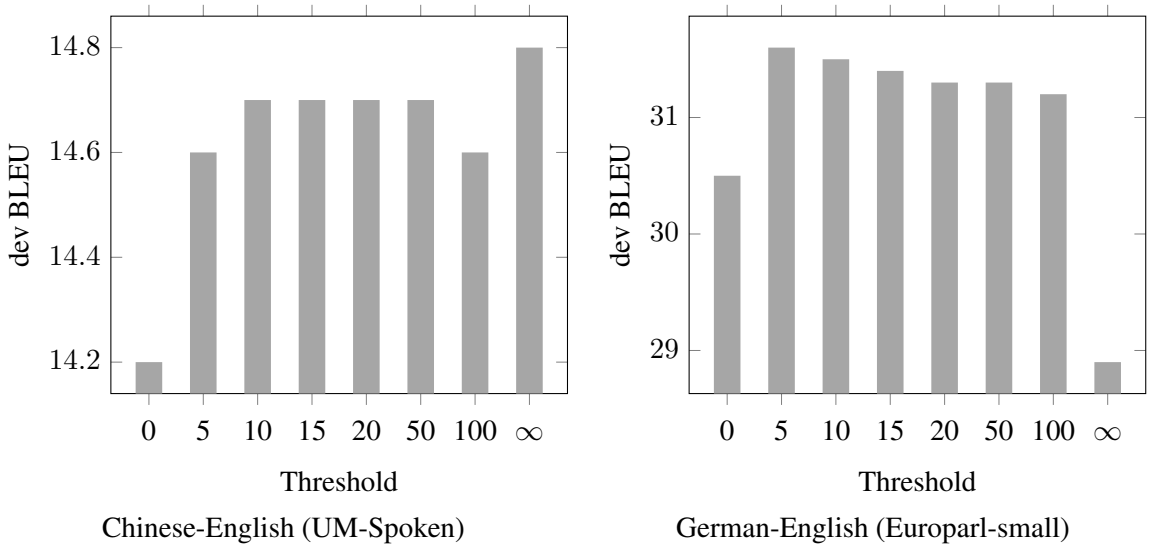


Figure 2: Effect on dev BLEU scores of the frequency threshold below which we fuse a word and attach its definition. These scores are used to choose the threshold that is used in Table 5.

3.6 Monolingual dictionaries

Because our dictionary-attachment method does not make any assumptions about the form of the definitions, we can apply it to monolingual source-language dictionaries as well. Monolingual source-language dictionaries are a natural resource for human translators, but we're not aware of previous research that uses them in data-driven machine translation. For many languages and language pairs, we expect them to be much more comprehensive than bilingual dictionaries. Our monolingual dictionary is the 汉语辞海 (*Hànyǔ Cíhǎi*),[10] which has a total of 380,579 entries. We removed pronunciations and concatenated multiple definitions into a single line. We did not resolve any cross-references in this dictionary, and we removed all entries with empty definitions. This gives us a final dictionary size of 358,234 entries.

We experimented with using this dictionary on the Spoken and Science UM datasets. The results are shown in Table 6. Although, as expected, it does not help as much as a bilingual dictionary, it does help on three out of four tasks we tried. All differences in this table are statistically significant.

		Test BLEU	
Segmentation	Dictionary	Spoken	Science
word	none	13.6	8.0
	zh–zh	14.3	8.4
	zh–en	15.4	9.2
BPE	none	16.6	11.2
	zh–zh	15.2	11.6
	zh–en	17.0	14.7

Table 6: Attaching a monolingual Chinese-Chinese dictionary improves over the baseline in three out of four tasks, although not as much as a bilingual Chinese-English dictionary does. All differences are statistically significant.

4 Analysis

To further examine how our methods improve translation, we looked at some examples in our UM-Spoken dev set, shown in Table 7 (word-based) and Table 8 (BPE). The (UNK) tag next to dictionary definitions indicates that the word is outside of the system's vocabulary.

In the first example, 对称性 (*duìchènxìng*, symmetry) is unknown to the word-based systems. Adding the definition to the parallel training data (*Append*) does not help word-based translation because the word remains unknown, whereas our model correctly generates the translation *symmetry*. With BPE, the word is broken into three pieces, so that the Append system can correctly generate the word *symmetry*. But the third character (性, *xìng*) can also mean "sex," and together with the following character (性感, *xìnggǎn*) can mean "sexy." This explains why the Append system incorrectly adds the words *of sex*.

In the second example, 火药 (*huǒyào*, gunpow-

[10] http://download.huzheng.org/zh_CN/

Source	1. 不 只 是 科学家们 对 对称性(UNK) 感 兴趣 。 2. 我 哥哥 听说 我们 做 了 火药(UNK) 。 3. 有些 登山者 经过 他 身旁 ， 打量(UNK) 了 他 一 番
Definitions	1. 对称性: symmetry 2. 火药: gunpowder(UNK) 3. 打量: to size sb(UNK) up to look sb(UNK) up and down to take the measure of to suppose to reckon
Reference	1. But it's not just scientists who are interested in symmetry. 2. Well, my brother heard that we had made gunpowder. 3. Some climbers had come by and looked at him,
Baseline	1. not only scientists are interested in the UNK of UNK. 2. My brother has heard that we've done a lot of work. 3. And some of the climber went to him, and he said,
Append	1. It's not just about scientists who are interested in UNK. 2. My brother has heard that we've done a lot of work. 3. And some of the UNK came over and over and over again,
Attach	1. not just scientists are interested in symmetry. 2. My brother heard that we had done UNK. 3. Some of the climber passed him, looked at him,

Table 7: Examples from word-based systems on the UM-Spoken data. In the first and second examples, the unknown words 对称性 (*duìchènxìng*) and 火药 (*huǒyào*) cannot be translated by the baseline, even with the dictionary in the parallel data (Append). Our model successfully incorporates the dictionary definition *symmetry*, but not *gunpowder*, because it is unknown. In the third example, the definition is not suitable as a direct translation of the unknown word 打量 (*dǎliàng*), but our model generates the word *looked*, apparently by picking out the word *look* from the definition and inflecting it correctly for the context.

BPE Source	1. 不 只 是 科学家们 对 对@@ 称@@ 性 感 兴趣。 2. 我 哥哥 听说 我们 做 了 火@@ 药 。 3. 有些 登@@ 山@@ 者 经过 他 身@@ 旁 ， 打@@ 量 了 他 一 番
Fused source	1. 不 只 是 科学家们 对 对称性(UNK) 感 兴趣 。 2. 我 哥哥 听说 我们 做 了 火药(UNK) 。 3. 有些 登山@@ 者 经过 他 身@@ 旁 ， 打量(UNK) 了 他 一 番 ,
Definitions	1. 对称性: sym@@ metry 2. 火药: gun@@ powder 3. 打量: to size s@@ b up to look s@@ b up and down to take the measure of to suppose to reck@@ on
Reference	1. But it's not just scientists who are interested in symmetry. 2. Well, my brother heard that we had made gunpowder. 3. Some climbers had come by and looked at him,
Baseline	1. not just scientists are interested in the sense of sympathy. 2. My brother had heard that we did a fire pills. 3. Some of the climbers passed him on the side, and he had a lot of money,
Append	1. Not only scientists are interested in the symmetry of sex. 2. My brother told us that we had done a fire. 3. Some of the climber passed his feet, and he took a second,
Fuse	1. not only scientists are interested in their interests in the world. 2. My brother has heard that we've done a good job. 3. Some of the climbers passed by him, and he gave him a sense,
Attach	1. not only scientists are interested in symmetry. 2. My brother heard that we did the gunpowder. 3. Some climbers passed by his side and looked at him,

Table 8: Examples from BPE-based systems on the UM-Spoken data. In the first two examples, the baseline system, even with the dictionary in the parallel data (Append), tries to translate the pieces of unknown words separately and incorrectly (e.g., *fire*, *pills*, *sex*). Our model is able to translate the first and third examples correctly as in Table 7, as well as the second example.

der) is unknown, and the definition word *gunpowder* is also unknown. So none of the systems are able to translate this word correctly (though arguably our system's generation of UNK is preferable). When we switch to BPE, our model generates the correct translation. The other systems fail because this word splits into two very common words, 火 (*hǔo*, fire), and 药 (*yào*, medicine), which the system tries to translate separately.

The third example shows what happens when we have a long definition that contains useful information, but is not suitable as a direct translation of the unknown word 打量 (*dǎliàng*). Here we see that our attachment model generates the word *looked*, apparently by picking out the word *look* from the definition and inflecting it correctly for the context. No other models were able to generate a word with a similar meaning.

Please see Appendix A for visualizations of the encoder-decoder attention for these three examples.

We also looked at a few examples from the Europarl-small dev set, shown in Table 9 and 10. In the first example, the definition *omission* was out of vocabulary, so our model was not able to perform any better than the baselines. However, in the BPE systems, our model was able to properly translate *Auslassung* to *omission* while none of the other baseline systems was able to. In the second example, we see something similar in the word-based system. The Baseline and Append models were unable to generate the correct translation of *Alternativlösung*, but our method was. With BPE, all systems (even Baseline) were able to translate the word correctly.

5 Discussion

In Section 1, we mentioned several other methods for using dictionaries in NMT, all of which treat dictionary definitions as target-language text. An alternative approach to handling rare words, which avoids dictionaries altogether, is to use word embeddings trained on large amounts of monolingual data, like fastText embeddings (Bojanowski et al., 2017). Qi et al. (2018) find that fastText embeddings can improve NMT, but there is a sweet spot (likely between 5k and 200k lines) where they have the most impact. They also find that pre-trained embeddings are more effective when the source and target languages are similar.

We, too, experimented with using fastText word embeddings in our NMT system, but have not seen any improvements over the baseline – perhaps because our datasets are somewhat larger than those used by Qi et al. (2018). We also experimented with using dictionaries to improve word embeddings and found that the present approach, which gives the model direct access to dictionary definitions, is far more effective.

The most significant limitation of our method is runtime: because it increases the length of the source sentences, training and decoding take 2–3 times longer. Another limitation is that the effectiveness of this method depends on the quality and coverage of the dictionaries.

In the future, we plan to experiment with additional resources, like thesauruses, gazetteers, or bilingual dictionaries with a different target language. Second, from our examples, we see that our model is able to select a snippet of the definition and adapt it to the target context (for example, by inflecting words), but further analysis is required to understand how much the model is able to do this. Finally, our method currently requires an exact match between a dictionary headword and a source word; we plan to extend the model to enable matching of headwords with inflected forms.

6 Conclusion

In this paper, we presented a simple yet effective way to incorporate dictionaries into a Transformer NMT system, by attaching definitions to source sentences to form a nonlinear structure that the Transformer can learn how to use. We showed that our method can beat baselines significantly, by up to 3.1 BLEU. We also analyzed our system's outputs and found that our model is learning to select and adapt parts of the definition, which it does not learn to do when the dictionary is simply appended to the training data. We also found that our method has some potential to work with monolingual dictionaries.

Acknowledgements

This paper is based upon work supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via contract #FA8650-17-C-9116. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, or the U.S. Government. The U.S. Gov-

Source	1. Ich hoffe , dass diese Auslassung(UNK) korrigiert werden kann . 2. Wäre das nicht eine Alternativlösung(UNK) ?
Definitions	1. Auslassung: omission(UNK) 2. Alternativlösung: alternative solution
Reference	1. I hope that this omission can be corrected. 2. Would this not be an alternative solution?
Baseline	1. I hope that this UNK can be corrected. 2. Would this not be a UNK?
Append	1. I hope that this UNK can be corrected. 2. Would this not be a UNK?
Attach	1. I hope that this UNK can be corrected. 2. Would this not be an alternative solution?

Table 9: Examples from word-based systems run on the Europarl-small data. In the first example, the dictionary defines unknown word *Auslassung* with another unknown word, *omission*, so neither adding the dictionary to the parallel data (Append) nor our model (Attach) benefits. In the second example, adding the dictionary definition of *Alternativlösung* to the parallel data does not help, but our model is able to incorporate it.

BPE source	1. Ich hoffe , dass diese Aus@@ l@@ assung korrigi@@ ert werden kann . 2. W@@ äre das nicht eine Altern@@ ativ@@ lösung ?
Fused source	1. Ich hoffe , dass diese Auslassung(UNK) korrigi@@ ert werden kann . 2. W@@ äre das nicht eine Alternativlösung(UNK) ?
Definitions	1. Auslassung: om@@ is@@ sion 2. Alternativlösung: alternative solution
Reference	1. I hope that this omission can be corrected. 2. Would this not be an alternative solution?
Baseline	1. I hope that this approval can be corrected. 2. Would this not be a alternative solution?
Append	1. I hope that this interpretation can be corrected. 2. Would this not be a alternative solution?
Fuse	1. I hope that these pieces can be corrected. 2. Would this not be a pronounce?
Attach	1. I hope that this omission can be corrected. 2. Would this not be an alternative solution?

Table 10: Examples from BPE-based systems run on the Europarl-small data. In the first exapmle, unlike in Table 9, the unknown word *Auslassung* is not replaced with UNK but is split into subwords, which the baseline system as well as the system with the dictionary in its parallel data (Append) translate incorrectly. Our model successfully uses the dictionary definition, *omission*. In the second example, BPE enables all models to translate the compound *Alternatvlösung* correctly.

ernment is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

References

Philip Arthur, Graham Neubig, and Satoshi Nakamura. 2016. Incorporating discrete translation lexicons into neural machine translation. In *Proc. EMNLP*, pages 1557–1567.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Trans. ACL*, 5:135–146.

Pi-Chuan Chang, Michel Galley, and Christopher D. Manning. 2008. Optimizing Chinese word segmentation for machine translation performance. In *Proc. WMT*, pages 224–232.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional sequence to sequence learning. In *Proc. ICML*, pages 1243–1252.

Mika Hämäläinen and Khalid Alnajjar. 2019. A template based approach for training NMT for low-resource Uralic languages - a pilot with Finnish. In *Proc. 2nd International Conference on Algorithms, Computing and Artificial Intelligence (ACAI)*, pages 520–525.

Michael McCloskey and Neal J. Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. *Psychology of Learning and Motivation*, 24:109–165.

Matt Post and David Vilar. 2018. Fast lexically constrained decoding with dynamic beam allocation for neural machine translation. In *Proc. NAACL HLT*, pages 1314–1324.

Ye Qi, Devendra Sachan, Matthieu Felix, Sarguna Padmanabhan, and Graham Neubig. 2018. When and why are pre-trained word embeddings useful for neural machine translation? In *Proc. NAACL HLT*, pages 529–535.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proc. ACL*, pages 1715–1725.

Liang Tian, Derek F. Wong, Lidia S. Chao, Paulo Quaresma, Francisco Oliveira, Yi Lu, Shuo Li, Yiming Wang, and Longyue Wang. 2014. UM-corpus: A large English-Chinese parallel corpus for statistical machine translation. In *Proc. LREC*, pages 1837–1842.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, pages 5998–6008.

Jiajun Zhang and Chengqing Zong. 2016. Bridging neural machine translation and bilingual dictionaries. arXiv:1610.07272.

A Attention Visualizations

Figures 3 and 4 show visualizations of the attention of our Attach model. They show the first layer of encoder-decoder attention when translating the three Chinese sentences of Tables 7 and 8. Note the translations are not exactly the same as shown above, because we used a beam size of one instead of the default of four.

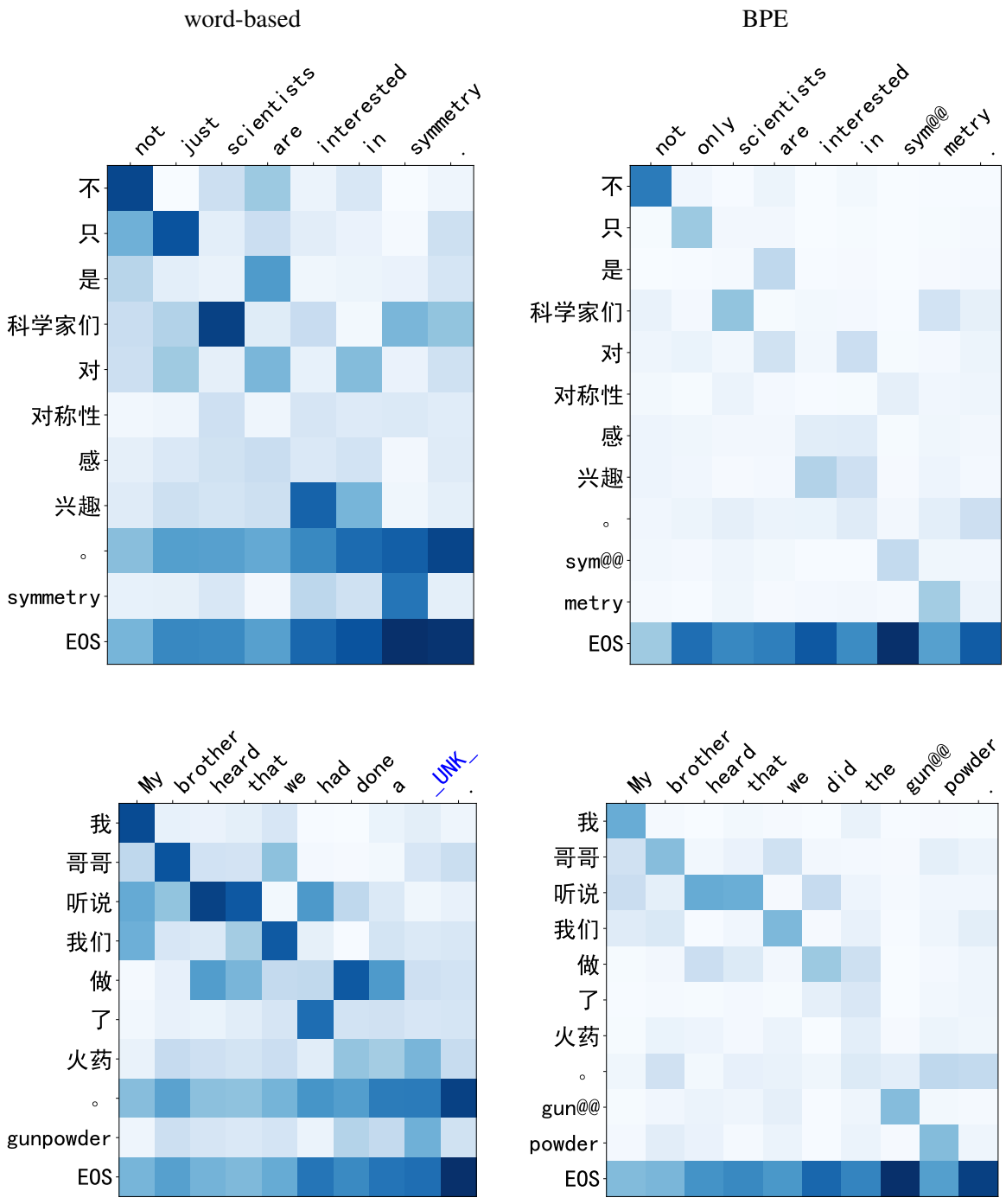


Figure 3: Attention visualizations for the first two Chinese-English examples of Tables 7 and 8.

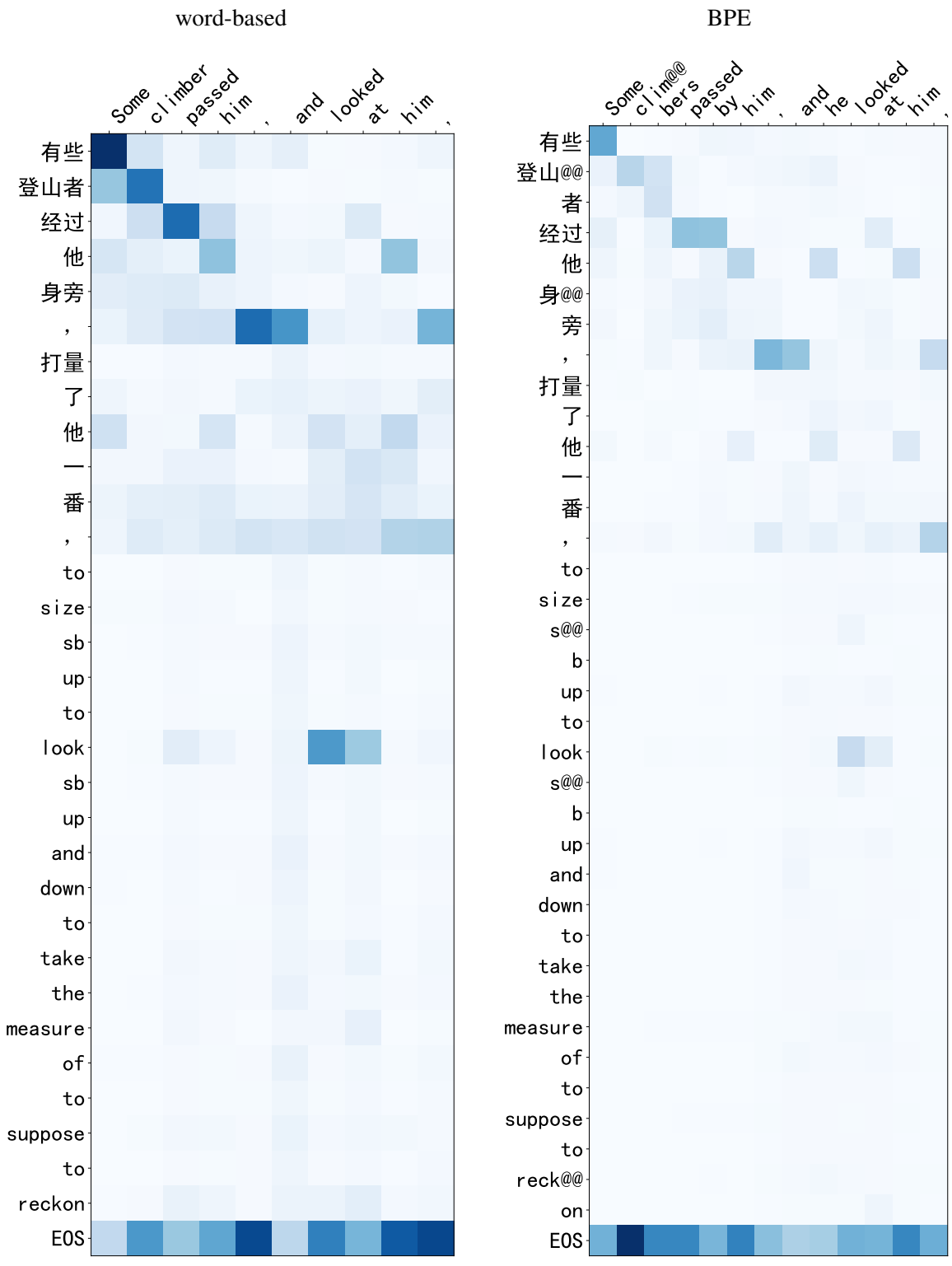


Figure 4: Attention visualizations for the third Chinese-English example of Tables 7 and 8.

Complete Multilingual Neural Machine Translation

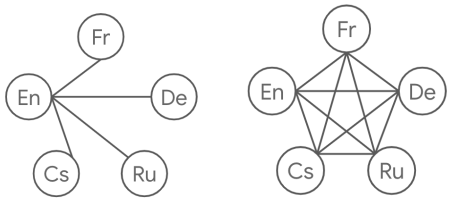
Markus Freitag and Orhan Firat
Google Research
{freitag,orhanf}@google.com

Abstract

Multilingual Neural Machine Translation (MNMT) models are commonly trained on a joint set of bilingual corpora which is acutely English-centric (i.e. English either as the source or target language). While direct data between two languages that are non-English is explicitly available at times, its use is not common. In this paper, we first take a step back and look at the commonly used bilingual corpora (WMT), and resurface the existence and importance of implicit structure that existed in it: multi-way alignment across examples (the same sentence in more than two languages). We set out to study the use of multi-way aligned examples to enrich the original English-centric parallel corpora. We reintroduce this direct parallel data from multi-way aligned corpora between all source and target languages. By doing so, the English-centric graph expands into a complete graph, every language pair being connected. We call MNMT with such connectivity pattern complete Multilingual Neural Machine Translation (cMNMT) and demonstrate its utility and efficacy with a series of experiments and analysis. In combination with a novel training data sampling strategy that is conditioned on the target language only, cMNMT yields competitive translation quality for all language pairs. We further study the size effect of multi-way aligned data, its transfer learning capabilities and how it eases adding a new language in MNMT. Finally, we stress test cMNMT at scale and demonstrate that we can train a cMNMT model with up to 111∗112=12,432 language pairs that provides competitive translation quality for all language pairs.

1 Introduction

Multilingual machine translation (Dong et al., 2015; Firat et al., 2016a; Johnson et al., 2017; Aharoni et al., 2019), which can serve multiple language pairs with a single model, has attracted much attention. In contrast to bilingual MT systems which can only serve one single language pair, multilingual models can serve $O(N^2)$ language pairs (N being the number of languages in a multilingual model) (Zhang et al., 2020).



(a) English-centric (b) Complete

Figure 1: Source-target translation graphs in MNMT. Solid lines indicate that there exist direct parallel data. When there is no line connecting any two languages, zero-resource or zero-shot approaches are employed.

The amount of available training data can differ a lot across language pairs and the majority of available MT training data is English-centric (Tiedemann, 2018; Arivazhagan et al., 2019b) which in practice means that most non-English language pairs do not see a single training example when training multilingual models (see Figure 1a). As a consequence, the actual performance of language pairs that do not include English on the source or target side lags behind the ones with large amounts of training data. Further, when increasing the number of languages, it gets (a) impractical to gather training data for each language pair and (b) challenging to find the right mix during training. Which is why models tasked with direct translation between non-English pairs either resort to bridging (pivoting) through a pivot language (Habash and Hu, 2009), or make use of synthetic parallel data (via back-translation) (Firat et al., 2016b; Chen et al., 2017) or study the problem under zero-shot settings (Johnson et al., 2017; Ha et al., 2016).

In this study, we make use of the potential pre-

Proceedings of the 5th Conference on Machine Translation (WMT), pages 550–560
Online, November 19–20, 2020.

existing multi-way property in the training corpora and generate as many direct training examples from pre-existing English-centric training data. If we can find training examples for each language pair in a multilingual mix, we call this model complete Multilingual Neural Machine Translation (cMNMT). cMNMT is then trained on all bilingual pairs between source and target languages by utilizing multi-way aligned training examples that consist of translations of the same sentence into multiple languages. We resurface multi-way aligned training examples by aligning training examples from different language pairs when either their source or target sides are identical (ie. pivoting through English, for German→English and English→French to extract German–French–English examples).

To make use of this data, the model samples a source and target language from the set of multi-way aligned corpus during training, which allows the model to see language pairs where originally no training data existed (missing connections in Figure 1a). As our experiments support, this method enables us to get access to training data for all tested language pairs (generating a complete graph (Figure 1b)). We will show that it is possible to generate a complete graph for at least a 6-language WMT setup. Some of the WMT training data is multi-way parallel by construction. Nevertheless, we show that we also find many training examples where the source and target origin from different sources. We further show on our 112 languages internal dataset, that we can find sufficient training data for over 12,000 language pairs by only providing 111 English-centric training corpora. This result indicates that it is possible to generate direct training data for many language pairs without the need for crawling new training examples. Our experiments suggest that before falling back to methods like zero-shot translation, you should investigate the structure of your pre-existing training data.

To address the problem of finding the right mix of examples from different language pairs during training, we further introduce a hierarchical sampling strategy that is language-specific (as opposed to being language pair specific). In addition to fixing some chronic issues of MNMT (i.e. low quality for out of English translation (Firat et al., 2016a; Johnson et al., 2017; Arivazhagan et al., 2019b)), the proposed sampling strategy efficiently ensures all source-target pairs are covered.

Experiments demonstrate that we can train a cMNMT model on a 30-language-pair WMT setup that outperforms bilingual and multilingual baselines as well as bridging on all non-English language pairs. We further show that the performance of the English language pairs stay stable and do not suffer from the changes in both the training data and the new training data sampling strategy. Furthermore, we share experiments at scale by demonstrating that we can train a cMNMT model that can serve 12,432 language pairs.

Our contribution is three-fold:

- We show that we can find a lot of training examples for all language pairs in a multilingual mix by only pivoting pre-existing English-centric training data. We further show that many of the extracted examples originate from different data sources and this method could scale to many more datasets. We also support these findings with experiments on our internal dataset, where we were able to find training data for all 12,432 language pairs.
- We demonstrate that cMNMT outperforms bilingual baselines, multilingual baselines as well as bridging on all non-English language pairs while keeping translation performance on English-centric language pairs.
- We introduce a new sampling strategy that is purely based on the target language instead of language pairs and does scale to MNMT models which hundreds of languages.

2 A Peek at Multi-way Aligned Examples in Bilingual Corpora

We choose six languages Czech (cs), English (en), French (fr), German (de), Spanish (es) and Russian (ru) from the public WMT datasets. The selection of the languages was driven by the fact that the WMT 2013 evaluation campaign (Bojar et al., 2013) released a multi-way test set for these six languages. As training data, we used WMT 2013 for Spanish, WMT 2014 for German, WMT 2015 for French, and WMT 2018 for Czech and Russian.

We can construct non-English bilingual training examples by pairing the non-English sides of two training examples with identical English translations. Table 1 shows the number of bilingual training examples that we could potentially extract from the English-centric training data. The

number of training examples for each non-English language pair varies from at least 0.3 million (Russian-German) to up to 4.8 million sentence pairs (Russian-French).

	cs	de	en	es	fr	ru
cs		0.7	47	0.8	1	0.9
de	0.7		4.5	2.3	2.5	0.3
en	47	4.5		13.1	38.1	33.5
es	0.8	2.3	13.1		10	4.4
fr	1	2.5	38.1	10		4.8
ru	0.9	0.3	33.5	4.4	4.8	

Table 1: WMT: Available training data (in million) after constructing non-English examples from English-centric examples with identical English side.

Some of the extracted non-English training examples are multi-way parallel by construction. The UN corpus is a 6-way parallel corpus, and three of the languages (English, French and Spanish) are in our 6-language mix. A portion of the Europarl corpus is again multi-way aligned. Nevertheless, a good amount of the extracted data is coming from different sources. Table 2 shows the number of non-English bilingual training examples separated by the two sources they originated from.

Table 3 shows how many translations are available for each sentence in the WMT training data. The majority (123 million) of the multi-way aligned examples do only have translations into two languages. As our original bilingual training data is English-centric, all of the 123 million training examples consist of an English sentence and a translation into one of our five other languages. A total of 13 million multi-way aligned examples are available in at least three languages. Further, Figure 2 shows the average number of translations conditioned by the language. Both Spanish and German have, on average more than three translations. In comparison, the majority of the multi-way aligned examples with Czech or English on the target side are bilingual (having only two translations). Our study resurfaced the inherent multi-way aligned information in the commonly used set of parallel corpora instead of discarding this information.

3 Complete Multilingual NMT

We call MNMT models that are trained for all possible source–target pairs as complete MNMT as all languages are connected via training data (also see Figure 1). Before going into details of how the missing pairs' data gathered, we recap MNMT first.

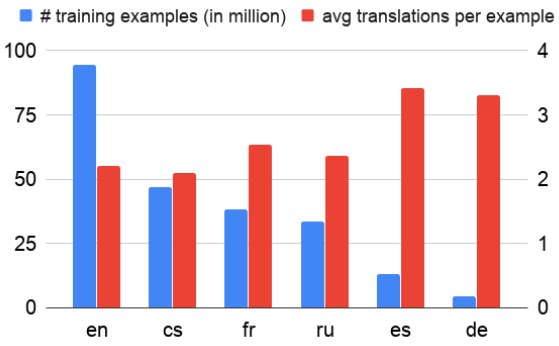


Figure 2: Average translations per multi-way aligned example conditioned on the target language.

Multilingual NMT Framework MNMT (Firat et al., 2016b; Johnson et al., 2017) is an extension of bilingual NMT which uses a single model to translate between multiple languages. The model parameters are trained on a joint set of bilingual corpora from different language pairs. Given the data imbalance across the different corpora, it is common to oversample the language pairs with less training data (Lee et al., 2016; Johnson et al., 2017). For a given language pair p, let $D(p)$ be the size of the available parallel corpus, the sample probability with a temperature T is defined as

$$p_p = \left(\frac{D(p)}{\sum_q D(q)}\right)^{\frac{1}{T}} \quad (1)$$

As a result, $T = 1$ corresponds to the actual data distribution, and $T = 100$ corresponds to (almost) an equal number of samples for each language pair). In addition to being able to translate language pairs that the model was trained with, the model can also translate between language pairs never seen explicitly during training which is often referred as zero-shot translation (Johnson et al., 2017; Ha et al., 2016).

Using multi-way aligned data in MNMT Instead of only relying on bilingual corpora, bilingual examples from different language pairs with identical target sentences can be combined into a single multi-way aligned training example. An example is given in Table 4. By comparing the English sides of the Spanish–English and the German–English corpora, we extract a multi-way aligned example that contains translations into all three languages.

	cc	CzEng	epps	nc	10^9	Paracrawl	UN	Wiki Titles	Yandex
Common Crawl (cc)	2.5M	13K	213	21k	10k	47k	1.8k	1	6.1k
CzEng 1.7		0	20k	417k	242k	55k	63k	98k	7.7k
Europarl (epps)			6.9M	1.2k	3.7k	4.8k	4.7k	255	280
News-Commentary (nc)				640k	186k	244	305	60	1.7k
10^9					0	12k	97k	1.5k	2.9k
Paracrawl						352k	18k	5.5k	1k
UN							16M	3.7k	3.3k
Wiki Titles								118k	4

Table 2: Number of training examples with identical English sides split by data sources. E.g. cell cc-CzEng shows the number of training examples with identical English side by only considering training data coming from either commoncrawl or CzEng for all language pairs (if available).

# languages	2	3	4	5	6
training data	123M	6.9M	5.4M	0.7M	10k

Table 3: Data statistics for the extracted multi-way aligned training examples for WMT: 123 million sentences are only available in 2 languages, while 10,000 sentences have translations in all 6 languages.

X-Y	*Bleib sicher* ↔ Stay safe
Z-Y	Mantente segura ↔ Stay safe
X-Y-Z	*Bleib sicher* ↔ Mantente segura ↔ Stay safe

Table 4: The two German–English and Spanish–English bilingual training examples can be combined into one multi-way aligned training example that consists of translations into all three languages.

While we can extract direct training data for any source-target pair among the languages considered, the total number of language pairs increases quadratically. The vanilla language pair based sampling strategy in Eq. (1) with adjustable temperature is capable of balancing low-high resource language pairs during training. However, we noticed a critical failure mode, which is further amplified in complete MNMT. The language-pair based sampling strategy (regardless of the temperature being used) *over-represents English* in English-centric models. Notice half of the languages have English on the source side, with the other half on the target side. This over-representation yields a schedule of examples for the encoder (resp. for the decoder) to see English examples half of the time throughout the entire training process. As a result, trained models end up favouring English either on the source and/or target. Although the implications on the encoder could be minimal, over-exposing English examples to the decoder curtail the learning signal when the target language is non-English. We hypothesise that this imbalance in the learning signal with respect to the target language is one of the roots of poor translation quality of multilingual models when translating out of English (Firat et al., 2016a; Johnson et al., 2017; Arivazhagan et al., 2019b).

To alleviate the *over-representation of English* with the language-pair based sampling strategy, we propose a hierarchical sampling strategy with two levels: i) we choose a target language (based on a temperature-based schedule), ii) uniformly sample a source language. Formally, for a given target language l, let $D(l)$ be the size of the available training examples with target language l, the sample probability with a temperature T is defined as

$$p_l = \left(\frac{D(l)}{\sum_q D(q)}\right)^{\frac{1}{T}} \quad (2)$$

During training, the scheduler samples a batch of training examples based on the target language only, as opposed to source-target language pair specific sampling. After choosing a target language, for each multi-way aligned example, we randomly (uniformly) pick one of the translations as the source sentence.

4 Experiments

We use a public transformer implementation with the transformer-big model size (Vaswani et al., 2017) for all multilingual setups. All bilingual models use a vocabulary of 32,000 subwords, while all multilingual models use a vocabulary of 64,000 subword units. All multilingual models are trained for 500,000 updates using an average batch size of around 33,000 sentences (∼1 million tokens). All bilingual models are trained for 400,000 steps as they converged earlier using a batch size of around 8,000 sentences (∼260,000 tokens). Due to the data imbalance across languages, we use a

temperature-based data sampling strategy to oversample low-resource language pairs in standard MNMT models (Equation 1) and low-resource target languages in cMNMT models (Equation 2). We use a temperature of $T = 5$ in both cases. All multilingual models add a token at the beginning of the input sentence to specify the required target language. All BLEU (Papineni et al., 2002) scores are calculated with sacreBLEU (Post, 2018).[1]

4.1 Baselines on WMT

We train several baselines: (i) bilingual models, (ii) multilingual models based on English-centric data, and (iii) bridging non-English language pairs.

Bilingual Baselines We train two bilingual baselines (using either transformer-base or transformer-big) for each language pair. In addition to training baselines on the original English-centric WMT data, we also train models for non-English language pairs on the extracted direct data (see Table 1). We experimented with several dropout rates for both setups and found that dropout=0.1 works best for transformer-base while dropout=0.3 works best for transformer-big. As can be seen from Table 5 and Table 6, the experiments suggest that the translation quality of the non-English language pairs is far behind the ones for English-centric language pairs. As an example, the translation quality between German and Russian reaches 6-7 BLEU only.

source \ target	cs	de	en	es	fr	ru
cs		16.6	30.4	20.7	22.6	13.9
de	15.2		29.5	27.0	28.7	6.9
en	25.2	25.7		33.6	34.8	23.6
es	15.6	22.7	33.9		34.2	18.7
fr	15.4	22.1	33.0	31.8		17.9
ru	12.5	6.1	28.4	22.6	24.3	

Table 5: BLEU scores on newstest2013 of bilingual models trained with the transformer-base architecture.

Multilingual Baselines We train a multilingual NMT model on the original WMT English-centric training data. BLEU scores are summarized in Table 7. All language pairs with English as the source or target language perform comparably well from at least 24.5 BLEU (English→Russian) up to 34.9 BLEU (English→French). The BLEU scores of non-English language pairs are consistently lower (which can be explained as a lack of supervision during training) and can be as low as 4.1 BLEU for Spanish→Czech or as high as 24.4 BLEU for French→Spanish.

source \ target	cs	de	en	es	fr	ru
cs		14.6	31.9	19.0	20.0	14.1
de	14.1		31.3	26.4	28.8	4.7
en	26.5	27.0		34.2	35.9	25.0
es	14.4	22.8	34.5		34.8	19.9
fr	13.0	20.7	34.2	32.5		18.6
ru	12.8	4.0	30.8	23.1	24.8	

Table 6: BLEU scores on newstest2013 of bilingual models trained with the transformer-big architecture.

source \ target	cs	de	en	es	fr	ru
cs		19.8	31.2	21.6	20.2	8.5
de	6.8		31.8	17.8	21.2	4.5
en	25.5	26.7		34.0	34.9	24.5
es	4.1	8.8	34.7		19.6	9.5
fr	4.2	11.2	33.8	24.4		6.5
ru	4.8	10.4	29.5	19.9	9.6	

Table 7: BLEU scores on newstest2013 of a MNMT model trained on English-centric training data. All non-English language pairs are unseen during training and BLEU scores measure zero-shot performance.

Bridging (Pivoting) Baselines The quality of MNMT is still behind the one from bilingual baselines for most of the language pairs (comparing Table 6 and Table 7). Nevertheless, having a single NMT model for each language pair is impractical, especially when increasing the number of language pairs. An alternative approach is called bridging (Cohn and Lapata, 2007; Wu and Wang, 2007; Utiyama and Isahara, 2007). For the bridging approach, we compromise and train only English-centric models. To enable the translation between non-English language pairs, the source sentence cascades through the source→English and English→target systems to generate the target sentence. This simple process has several limitations: (i) translation errors accumulate in the pipeline, (ii) decoding time gets doubled since inference has to be run twice, (iii) bridging through a morphologically low language (i.e. English), important information could be lost (i.e. gender). The BLEU scores (Table 8) for all non-English pairs are higher compared to all previous baselines. We can reach acceptable translation quality even for

[1] sacreBLEU signatures: BLEU+case.mixed+lang.SRC-TGT+numrefs.1+smooth.exp+SET+tok.intl+version.1.2.20

German→Russian, where our direct training data is scarce. We use the bridging baseline to compare our cMNMT models in the rest of the paper.

source \ target	cs	de	en	es	fr	ru
cs		22.4	31.9	27.0	28.8	21.9
de	21.5		31.3	26.9	29.0	20.3
en	26.5	27.0		34.2	35.9	24.9
es	22.6	22.8	34.5		32.6	22.5
fr	21.4	22.2	34.2	29.1		21.6
ru	21.3	20.6	30.8	27.2	28.5	

Table 8: BLEU scores on newstest2013 for our WMT setup. Translations for non-English language pairs are generated via bridging over English.

source \ target	cs	de	en	es	fr	ru
cs		25.8 +3.4	32.0 +0.1	30.1 +3.1	31.4 +2.6	26.9 +5.0
de	23.9 +2.4		31.2 -0.1	29.9 +3.0	31.8 +2.8	23.4 +3.1
en	26.9 +0.4	27.1 +0.1		35.0 +0.8	35.5 -0.4	26.4 +1.5
es	24.9 +2.3	25.7 +2.9	34.9 +0.4		36.0 +3.4	24.9 +2.4
fr	23.7 +2.3	25.2 +3.0	34.2 +0.0	33.3 +4.2		23.5 +1.9
ru	24.3 +3.0	22.7 +2.1	29.8 -1.0	28.6 +1.4	30.1 +1.6	

Table 9: BLEU on newstest2013 for our novel cMNMT model. The small numbers are the difference (ΔBLEU) with respect to the bridging approach (Table 8).

4.2 Complete MNMT Models on WMT

Without adding new training data and taking into account the multi-way property of the data, we train a complete multilingual NMT system (cMNMT, see Section 3). We compare the performance of cMNMT with the best baseline model that is based on bridging (Table 8) and report BLEU and delta BLEU numbers in Table 9. The BLEU scores for the non-English language pairs go up from at least 1.4 BLEU for Russian→Spanish up to 5.0 BLEU for Czech→Russian. We changed the sampling strategy for our cMNMT models to be conditioned on the target language only (Section 3). As a result, English has been seen less often as the target language when compared to a standard MNMT setup. Interestingly, this seems to affect only the performance of Russian→English, which shows a decrease of 1 BLEU point. The other language pairs with English as the target language are keeping their translation quality.

When comparing our cMNMT model to the English-centric baseline (Table 7), we see an average BLEU increase of 14.6 BLEU for all non-English language pairs. It is worth noticing that every language pair has now at least 22 absolute BLEU points. Interestingly, the absolute BLEU scores in each row (translations into the same language) are much closer, suggesting a more universal input representation.

5 Analysis and Discussion

To further understand the impact of multi-way aligned examples on NMT, we run a couple of additional experiments.

Training Data Sampling Strategy In Section 3, we did introduce our new training data sampling strategy that is based on the target language only. This change was mainly driven by the fact that having a language-pair conditioned schedule is not scalable when building a system of 12,432 language pairs. Instead of finding a good sampling weight for each of the 12,432 language pairs, we only need to find a suitable mix for the 112 target languages. Further, we have more control over how often each target language will be seen during training. To see the impact of this change, we train an MNMT system on the joint set of the 30 different bilingual corpora with a standard language-pair based temperature scheduling scheme and compare it to a cMNMT model. We used temperature 5 in both setups. ΔBLEU numbers for each language-pair can be seen in Figure 3. The language-conditioned temperature scheduling increases BLEU scores for 29 out of 30 language-pairs with larger gains for the low-resource language-pairs. This experiment suggests that a target language based temperature scheduling is not only simpler but also performs better on average.

Separate Multi-way Aligned Examples We test the transfer learning capability of cMNMT by training a cMNMT model only on the 13 million multi-way aligned examples that have translations in at least three languages (see Table 3). In other words, we remove all training examples that are only available in English and one additional language. If no transfer learning is happening, the English-centric scores will decrease while the BLEU numbers of the non-English language

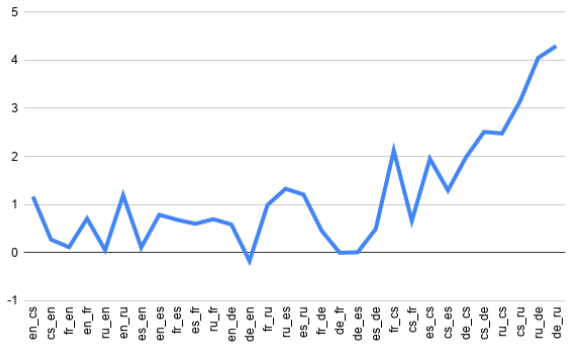


Figure 3: ΔBLEU scores for a target language based versus a language-pair based temperature schedule.

pairs are not affected. Experimental results can be seen in Table 10. Interestingly, we find that the performance of all language pairs is similarly affected. This indicates that transfer learning is happening between the language pairs and that non-English language pairs benefit from having more English-centric data.

source \ target	cs	de	en	es	fr	ru
cs		24.1 -1.7	30.3 -1.7	28.4 -1.7	29.4 -2.0	23.6 -3.3
de	19.1 -4.8		30.6 -0.6	29.1 -0.8	30.5 -1.3	21.7 -1.7
en	21.8 -5.1	26.2 -0.9		33.4 -1.6	34.6 -0.9	23.3 -3.1
es	19.7 -5.2	24.9 -0.8	34.3 -0.6		35.1 -0.9	22.7 -2.2
fr	18.6 -5.1	24.0 -1.2	33.0 -1.2	32.3 -1.0		21.3 -2.2
ru	19.7 -4.6	21.4 -1.3	27.7 -1.9	27.1 -1.5	27.9 -2.2	

Table 10: BLEU on newstest2013 for a model trained on 13 million multi-way aligned (n>2) data. Small numbers are the difference (ΔBLEU) between cMNMT trained on all multi-way examples (136M, Table 9).

To further study this effect, we reverse that experiment and remove all examples that have translations into more than two languages. This experiment investigates if the non-English language pairs in a standard MNMT model can benefit from having training examples with identical English sides. Experimental results can be found in Table 11. The BLEU scores for English-centric language pairs drop by 0.9 points on average while the performance of non-English language pairs decreases by 1.6 BLEU on average.

source \ target	cs	de	en	es	fr	ru
cs		17.8 -2.0	31.5 +0.3	14.5 -7.1	20.3 +0.1	5.2 -3.3
de	7.8 +1.0		29.6 -2.2	17.2 -0.6	22.9 +1.7	1.8 -2.7
en	25.6 +0.1	23.9 -2.8		33.1 -0.9	33.5 -1.4	24.6 +0.1
es	7.2 +3.0	3.9 -5.8	32.7 -1.1		24.3 +4.7	7.6 -1.9
fr	6.7 -2.5	13.0 -1.8	33.1 -0.7	19.3 -5.1		7.2 +0.7
ru	5.5 +0.7	10.4 +0.0	29.3 -0.2	8.5 -14.4	13.7 +4.1	

Table 11: BLEU on newstest2013 for a model trained on 2-way data only. Small numbers are the difference (ΔBLEU) between the vanilla MNMT model (Table 7).

Leave N-Out We further investigate the transfer learning capability of our approach by training several cMNMT models on different amounts of training data. We start with a cMNMT model trained on English-centric bilingual training data only. This setup ensures that all languages have been seen on both the source and target side during training. We further group the remaining multi-way aligned training examples by target language and add one after another to the training data. Important to mention: We retrained all configurations from scratch. Experimental results are summarized in Figure 4. We report average BLEU scores grouped by the target language. We can see that adding training data $x \rightarrow y$ for a target language y, gives a significant boost in translation quality for that target language. These results demonstrate that even though we can translate between language pairs without seeing a single example during training, adding supervision during training significantly increases BLEU scores.

Adding a New Language We further investigate how a cMNMT model behaves when fine-tuned (Freitag and Al-Onaizan, 2016) to a new language. We chose Italian as the new language as the test set newstest2009 is multi-way in Czech, English, French, German, Italian and Spanish and thus we can report BLEU scores between all language pairs. We run two experiments with two different sets of fine-tuning data. First, we fine-tuned the cMNMT model (Table 9) on English↔Italian news-commentary (45,000 examples). Second, we converted the same data into multi-way aligned examples by augmenting the bilingual examples with translations into other languages when found in our

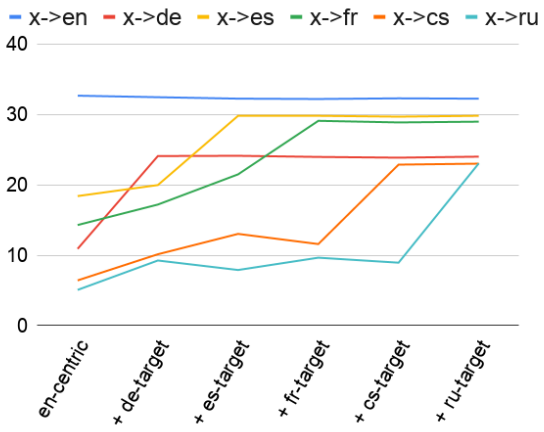


Figure 4: BLEU scores of models using only parts of the multi-way training data.

original training data. Experimental results for fine-tuning our model for one epoch on either of the two datasets can be found in Table 12. Both fine-tuning experiments show the same BLEU improvements for Italian↔English. Nevertheless, when only fine-tuning on English↔Italian data, we sacrifice translation quality for most of the language pairs which can be seen in the $x \rightarrow y$ column. Further, fine-tuning on multi-way aligned examples does improve the average BLEU scores by 4.3 BLEU for translations into Italian ($x \rightarrow$it). Overall, these experiments suggest that fine-tuning with multi-way aligned data is superior.

model	it→en	it→ x	en→it	$x \rightarrow$it	$x \rightarrow y$
cMNMT	13.5	9.7	2.3	2.6	22.0
+ft en↔it	21.5	14.2	13.6	11.8	17.8
+ft mway	21.2	18.5	13.5	11.9	23.0

Table 12: BLEU scores for newstest2009 for fine-tuning (ft) our cMNMT model on either English↔Italian (it↔en) news-commentary or on the same sentences but augmented with translations into other languages (mway), if available. Column $x \rightarrow y$ shows average BLEU scores for all language pairs.

Scaling cMNMT: 12,432 Language Pairs We run additional experiments on a 112 language in-house dataset (Arivazhagan et al., 2019b) to see if our approach scales to 12,432 language pairs. Our in-house dataset does not only contain more languages than the WMT setup, but also has a much wider range of available training resources. While for the high resource languages, we have access to billions of training examples, most of the low resource languages have less than 1 million training examples. We refer the reader to the description in Arivazhagan et al. (2019b) for more details regarding the dataset. Figure 5 shows the training data sizes and the average translations per multi-way example.

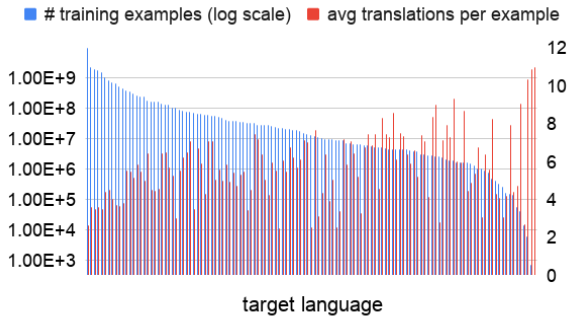


Figure 5: Average translations per multi-way example conditioned on the target language.

Although a deeper and wider architecture does improve the quality of multilingual models for this dataset, we use the same experimental setup as used in our WMT experiments (see Section 4) to run an MNMT and cMNMT model on our in-house data. Experimental results can be seen in Table 13. cMNMT outperforms MNMT for non-English languages by 10.1 BLEU points on average while keeping the translation quality for language pairs that include English as source or target. These results demonstrate that our proposed approach does scale far behind the six language WMT setup.

	en→ x	$x \rightarrow$en	$x \rightarrow y$
all	+0.34	-0.05	+10.1
low resource	+0.23	-0.15	+8.82
mid resource	+0.35	-0.05	+11.02
high resource	+0.45	+0.04	+9.73

Table 13: Average BLEU difference (ΔBLEU) between a cMNMT and a vanilla MNMT model for our in-house 112 language setup. Positive numbers present improvement of cMNMT over MNMT.

6 Related Work

Direct models To translate between languages with little training data, three general approaches emerged, i. bridging through a third language (pivot-based MT) (Cheng et al., 2016; Currey and Heafield, 2019), ii. generating pseudo-parallel data between direct language pairs and training the direct pairs with that (zero-resource MT) (Firat et al., 2016b; Chen et al., 2017) and, iii. zero-shot methods where the model is asked to translate a direct pair only at test time (Johnson et al., 2017; Ha et al., 2016; Arivazhagan et al., 2019a).

Although pivot-based approaches perform sufficiently good when cascaded with strong bilingual models (Gu et al., 2019), their practicality is limited due to compounding errors from pipelining and doubled inference cost. The zero-resource approaches, combined with iterative-back translation (Hoang et al., 2018) are quite powerful but their inefficiency is worth noting. For N languages, one needs to devise a training routine that could sample $N^2 - N$ pairs, generate pseudo-parallel data. The added time to generate pseudo-parallel data for every pair grows quadratically, making it challenging for systems considering a large number of languages. Recently, by devising a practical sub-sampling approach, (Zhang et al., 2020) demonstrated zero-resource techniques could be scaled to massively multilingual setup. We find the study by (Zhang et al., 2020) closest to our work, having the goal of any-to-any multilingual translation. But compared to sampling language pairs with no parallel data and generating pseudo-parallel data on-the-fly, our approach makes use of existing multi-way alignment information before training. Lastly, zero-shot approaches attempt to measure the generalization performance of the MNMT models, but to date, the zero-shot quality still trails behind the pivot and zero-resource methods (Al-Shedivat and Parikh, 2019). Our proposed cMNMT, naturally fills the gap between these three approaches, the multi-way data can be extracted offline, and efficiently be mixed with the original data using a hierarchical data sampler. It does not require extra steps to generate pseudo-parallel data, and (as expected) it handily outperforms zero-shot approaches.

N-way data In this paper, we only made use of multi-way aligned data to sample bilingual pairs out of it. But there exist several approaches that make use of the multi-view structure in the data, such as Dabre et al. (2019), who explored the use of small multi-parallel corpora a for one-to-many NMT. Another approach is multi-source NMT (Zoph and Knight, 2016). Although multi-source NMT is a promising direction, it has practical problems such as lacking multiple sources at inference time (Nishimura et al., 2018). We believe research in this direction will be the key to improve mid/high-resource NMT and address several robustness issues to the input noise. Aulamo et al. (2020) recently released MultiParaCrawl where the authors extracted direct data for non-English language pairs from the English-centric Paracrawl corpus.

Sampling scheduling Several approaches proposed to address data sampling for multi-task models, some relying on temperature-based heuristics (Lee et al., 2016; Devlin et al., 2018; Arivazhagan et al., 2019b), others relying on adaptive schedules that incorporate the model gains, baselines or quality expectations into the data schedulers (Kiperwasser and Ballesteros, 2018; Jean et al., 2019; Wang et al., 2020). We believe data sampling is a critical research area for not only MNMT but also multi-task learning in general. We reveal a critical failure mode of the commonly used temperature sampling strategy, and how it causes the poor translation quality while translating out of English.

7 Conclusion

In this work, we introduced complete Multilingual Neural Machine Translation (cMNMT) that exploits the multi-way alignment information in the underlying training data to improve translation quality for language pairs where training data is scared or not available. Standard MNMT models are trained on a joint set of different training corpora for a variety of language pairs. cMNMT combines the different corpora and constructs multi-way aligned training examples that consist of translations of the same sentence into multiple languages. In combination with a novel temperature-based sampling approach that is conditioned on the target language only, we show that cMNMT is superior to the standard MNMT model and the even better-performing bridging approach. Experimental results on a public WMT 30 language pairs dataset and an in-house 12,432 language pairs dataset demonstrated an average BLEU increase of more than 10 BLEU points for non-English language pairs. This approach leads to a single NMT model that can serve 12,432k language pairs with reasonable quality which also surpasses the translation quality of the bridging approach, which is nowadays used in most modern MT services.

References

Roee Aharoni, Melvin Johnson, and Orhan Firat. 2019. Massively Multilingual Neural Machine Translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,*

Volume 1 (Long and Short Papers), pages 3874–3884.

Maruan Al-Shedivat and Ankur Parikh. 2019. Consistency by agreement in zero-shot neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1184–1197.

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Roee Aharoni, Melvin Johnson, and Wolfgang Macherey. 2019a. The missing ingredient in zero-shot neural machine translation. *CoRR*, abs/1903.07091.

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George Foster, Colin Cherry, Wolfgang Macherey, Zhifeng Chen, and Yonghui Wu. 2019b. Massively multilingual neural machine translation in the wild: Findings and challenges. *CoRR*, abs/1907.05019.

Mikko Aulamo, Umut Sulubacak, Sami Virpioja, and Jörg Tiedemann. 2020. Opustools and parallel corpus diagnostics. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 3782–3789.

Ondřej Bojar, Christian Buck, Chris Callison-Burch, Christian Federmann, Barry Haddow, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. 2013. Findings of the 2013 Workshop on Statistical Machine Translation. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 1–44, Sofia, Bulgaria. Association for Computational Linguistics.

Yun Chen, Yang Liu, Yong Cheng, and Victor OK Li. 2017. A Teacher-Student Framework for Zero-Resource Neural Machine Translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1925–1935.

Yong Cheng, Yang Liu, Qian Yang, Maosong Sun, and Wei Xu. 2016. Neural machine translation with pivot languages. *CoRR*, abs/1611.04928.

Trevor Cohn and Mirella Lapata. 2007. Machine Translation by Triangulation: Making Effective Use of Multi-Parallel Corpora. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 728–735, Prague, Czech Republic. Association for Computational Linguistics.

Anna Currey and Kenneth Heafield. 2019. Zero-resource neural machine translation with monolingual pivot data. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 99–107.

Raj Dabre, Atsushi Fujita, and Chenhui Chu. 2019. Exploiting multilingualism through multistage fine-tuning for low-resource neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1410–1416.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. 2015. Multi-task learning for multiple language translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1723–1732, Beijing, China. Association for Computational Linguistics.

Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. 2016a. Multi-Way, Multilingual Neural Machine Translation with a Shared Attention Mechanism. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 866–875.

Orhan Firat, Baskaran Sankaran, Yaser Al-Onaizan, Fatos T Yarman Vural, and Kyunghyun Cho. 2016b. Zero-resource translation with multi-lingual neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 268–277.

Markus Freitag and Yaser Al-Onaizan. 2016. Fast domain adaptation for neural machine translation. *arXiv preprint arXiv:1612.06897*.

Jiatao Gu, Yong Wang, Kyunghyun Cho, and Victor O. K. Li. 2019. Improved zero-shot neural machine translation via ignoring spurious correlations. *CoRR*, abs/1906.01181.

Thanh-Le Ha, Jan Niehues, and Alexander Waibel. 2016. Toward multilingual neural machine translation with universal encoder and decoder. *arXiv preprint arXiv:1611.04798*.

Nizar Habash and Jun Hu. 2009. Improving Arabic-Chinese statistical machine translation using English as pivot language. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 173–181, Athens, Greece. Association for Computational Linguistics.

Vu Cong Duy Hoang, Philipp Koehn, Gholamreza Haffari, and Trevor Cohn. 2018. Iterative back-translation for neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 18–24, Melbourne, Australia. Association for Computational Linguistics.

Sébastien Jean, Orhan Firat, and Melvin Johnson. 2019. Adaptive scheduling for multi-task learning.

Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Eliyahu Kiperwasser and Miguel Ballesteros. 2018. Scheduled multi-task learning: From syntax to translation. *Transactions of the Association for Computational Linguistics*, 6:225–240.

Jason Lee, Kyunghyun Cho, and Thomas Hofmann. 2016. Fully character-level neural machine translation without explicit segmentation. *CoRR*, abs/1610.03017.

Yuta Nishimura, Katsuhito Sudoh, Graham Neubig, and Satoshi Nakamura. 2018. Multi-source neural machine translation with missing data. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 92–99, Melbourne, Australia. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Matt Post. 2018. A Call for Clarity in Reporting Bleu Scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191.

Jörg Tiedemann. 2018. Emerging language spaces learned from massively multilingual corpora. *CoRR*, abs/1802.00273.

Masao Utiyama and Hitoshi Isahara. 2007. A comparison of pivot methods for phrase-based statistical machine translation. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 484–491.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Xinyi Wang, Yulia Tsvetkov, and Graham Neubig. 2020. Balancing training for multilingual neural machine translation.

Hua Wu and Haifeng Wang. 2007. Pivot language approach for phrase-based statistical machine translation. *Machine Translation*, 21(3):165–181.

Biao Zhang, Philip Williams, Ivan Titov, and Rico Sennrich. 2020. Improving massively multilingual neural machine translation and zero-shot translation.

Barret Zoph and Kevin Knight. 2016. Multi-source neural translation. *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Paraphrase Generation as Zero-Shot Multilingual Translation: Disentangling Semantic Similarity from Lexical and Syntactic Diversity

Brian Thompson
Johns Hopkins University
brian.thompson@jhu.edu

Matt Post
Johns Hopkins University
post@cs.jhu.edu

Abstract

Recent work has shown that a multilingual neural machine translation (NMT) model can be used to judge how well a sentence paraphrases another sentence in the same language (Thompson and Post, 2020); however, attempting to *generate* paraphrases from such a model using standard beam search produces trivial copies or near copies. We introduce a simple paraphrase generation algorithm which discourages the production of n-grams that are present in the input. Our approach enables paraphrase generation in many languages from a single multilingual NMT model. Furthermore, the amount of lexical diversity between the input and output can be controlled at generation time. We conduct a human evaluation to compare our method to a paraphraser trained on the large English synthetic paraphrase database ParaBank 2 (Hu et al., 2019c) and find that our method produces paraphrases that better preserve meaning and are more gramatical, for the same level of lexical diversity. Additional smaller human assessments demonstrate our approach also works in two non-English languages.

1 Introduction

Paraphrase generation is the task of producing a fluent output sentence which is semantically similar to the input sentence while being syntactically and/or lexically different from it (Bhagat and Hovy, 2013). Paraphrasing has been of longstanding interest in the NLP community (McKeown, 1983) and has been used for data augmentation in question answering (Dong et al., 2017; Gan and Ng, 2019), machine translation (MT) (Hu et al., 2019a; Khayrallah et al., 2020), task oriented dialog (Niu and Bansal, 2018, 2019), and MT metrics (Banerjee and Lavie, 2005; Zhou et al., 2006; Denkowski and Lavie, 2010; Thompson and Post, 2020).

Thompson and Post (2020) recently released the Prism MT metric, which uses a multilingual neural MT (NMT) model as a paraphraser to *score* paraphrastic pairs; they treat paraphrasing as a zero-shot translation task (e.g., "translation" from English to English) and force-decode and score MT system outputs conditioned on their respective human translations. They denote their paraphraser as *lexically/syntactically unbiased* as it does *not* prefer output that differs lexically or syntactically from the input; this is advantageous for an MT metric as it assigns the highest score to an MT output which matches or nearly matches a human reference, but generating from the Prism model using standard beam search produces trivial copies or near copies.

We introduce a simple method to enable paraphrase generation from a multilingual NMT model.[1] Our method discourages the model from producing n-grams that match n-grams in the input sentence. This serves to lexically bias the output away from the input sentence, resulting in non-trivial paraphrases.

When considered together with Prism model of Thompson and Post (2020), our paraphrase generation approach offers several potential advantages over the common technique of training a paraphrase model on synthetic paraphrases generated by translating one side of bitext into the language of the other side (Wieting et al., 2017; Wieting and Gimpel, 2018; Hu et al., 2019c):

- The fluency/semantic similarity vs lexical diversity trade-off can be controlled at generation time.
- The approach works in many languages, with a single model.
- The approach addresses an inherent shortcoming in creating synthetic paraphrases from bi-

[1] We release our code at https://github.com/thompsonb/prism

Proceedings of the 5th Conference on Machine Translation (WMT), pages 561–570
Online, November 19–20, 2020.

text in which ambiguities in one language can create errorful synthetic paraphrases in the other (see §6).

- Separating the fluency and semantic similarity model from the lexical and/or syntactic diversity model allows them to be developed and evaluated with less interdependencies.

We conduct human evaluations to compare our proposed method to a strong English baseline paraphraser trained on the ParaBank 2 dataset (Hu et al., 2019c), which consists of 50 million synthetic examples generated by translating the Czech side of Czech–English bitext into English and pairing it with the original English. We find that our method outperforms this baseline—both in terms of semantic similarity and grammaticality—when our system is adjusted to match the lexical diversity of the baseline. We also present small scale evaluations that suggest our method is effective in other languages.

2 Related Work

Paraphrase Generation Machine translation techniques can be used to train paraphrase models (Quirk et al., 2004). Another method to generate a paraphrase is to translate a text to a different language and then back again (Mallinson et al., 2017). Multiple pivot languages can be used to lessen the effect of inherent ambiguities (Aziz and Specia, 2013), at the expense of complication. Several works have focused on training on paraphrase data, including synthetic data created by starting with bitext and translating one side into the language of the other side to create synthetic paraphrases (Wieting et al., 2017; Wieting and Gimpel, 2018; Hu et al., 2019c). Ideas such as adversarial training (Iyyer et al., 2018), reinforcement learning (Li et al., 2018), and variational autoencoders (Gupta et al., 2018; Chen et al., 2019b) have also been explored in the context of paraphrase generation.

Diversity in Generation Creating paraphrases which differ from their input in non-trivial ways is a challenging problem. Hu et al. (2019c) used constrained decoding (Hokamp and Liu, 2017) in conjunction with a set of constraints (e.g., avoiding certain words which are present in the input) when creating synthetic paraphrases from bitext. Kajiwara (2019) also used hard constraints, but at decoding time. Our work is similar but uses "soft" constraints (i.e., down-weighting tokens which complete n-grams in the input, but not disallowing them all together). Another approach is to control generation with syntactic examples (Iyyer et al., 2018; Chen et al., 2019a) or codes (Shu et al., 2019).

Multilingual NMT Multilingual NMT (Dong et al., 2015) has been shown to enable zero-shot translation—that is, translation between languages pairs not included in training (e.g., translating from Spanish→Arabic at test time when the model was trained on Spanish→English and English→Arabic, but not Spanish→Arabic) (Johnson et al., 2017; Gu et al., 2018; Pham et al., 2019). Zhou et al. (2019) also explored incorporating paraphrase data into training to improve multilingual NMT performance.

Tiedemann and Scherrer (2019) explored using paraphrase recognition to test the semantic abstraction of a fairly small multilingual NMT system trained on Bibles and also demonstrate the model's ability to paraphrase in English. However, they did not perform a human evaluation of paraphrase quality, and Thompson and Post (2020) found that simply generating via beam search from a multilingual NMT model trained on a large general domain corpus results in trivial copies most of the time. We build upon Tiedemann and Scherrer (2019) by using a larger, general domain model, introducing a novel generation algorithm to produce output with lexical diversity, and performing human evaluations.

Paraphrastic similarity Similarity between intermediate representations produced by multilingual NMT encoders has been used to measure semantic similarity and/or paraphrastic similarity (Schwenk and Douze, 2017; Wieting et al., 2019; Raganato et al., 2019). Similarly, Prism (Thompson and Post, 2020) use a multilingual NMT model as a lexically/syntactically unbiased paraphraser for scoring MT system outputs conditioned on their associated human reference translations. We build on this by introducing a lexical bias away from the input at generation time, enabling the use of a multilingual NMT model as a generative paraphraser.

3 Method

Let x and y be sentences, let $\mathcal{M}(x)$ represent the meaning of x, and let $S(x, y)$ measure the lexical and/or syntactic similarity between the two sentences. Formally, we can state the problem of para-

Algorithm 1 Before paraphrasing a sentence, `buildPenalties()` is called to construct a mapping of word prefixes to subwords that require penalties. Then, `penalize()` is called to modify the model prediction `targetLogProbs` at every decoder timestep.

```
def buildPenalties(source):
  penalties = defaultdict(list)
  for n in [1, 2, 3, 4]:
    for ngram of size n in subwords2words(source):
      prefix, word = ngram[0:-1], ngram[-1]
      for subword in targetVocab:
        if word.lower().startsWith(subword.lower()):
          penalties[prefix].append(subword)
  return penalties

def penalize(history, penalties, targetLogProbs):
  for n in [1, 2, 3, 4]:
    prefix = subwords2words(history)[-(n-1):]
    for subword in penalties[prefix]:
      targetLogProbs[id(subword)] -= alpha * (n ** beta)
```

phrase generation as finding $\hat{y}$:

$$\hat{y} = \underset{y}{\operatorname{argmax}} \left[p(y \mid \mathcal{M}(x)) - \alpha S(x, y) \right] \quad (1)$$

where α controls the semantic similarity and fluency vs lexical and/or syntactic diversity trade-off.

3.1 Lexically/Syntactically Unbiased Paraphraser

The intralingual probability $p(y \mid \mathcal{M}(x))$ can be viewed as a lexically/syntactically unbiased paraphraser. This model is responsible for producing output which is both semantically similar to the input and fluent, but has no notion of lexical and/or syntactic diversity. We use the multilingual NMT system released with Prism to model $p(y \mid \mathcal{M}(x))$.

3.2 Lexical Bias

We choose n-gram overlap as our measure of lexical and/or syntactic similarity $S(x, y)$, and propose a simple n-gram overlap measure that penalizes the production of n-grams matching n-grams in the input sequence to enable the paraphrase generation. Our proposed algorithm begins by constructing a set of all (word) n-grams, $1 \leq n \leq 4$, from the input.[2] At each decoding step, the algorithm checks whether any of the target vocabulary subwords *begin* the last word of an input n-gram.[3] All such subwords are penalized by subtracting αn^{β} from the output log probabilities of the NMT model before selecting candidates to extend the beam, where n is the n-gram length, α is the user-specified trade-off between semantic similarity and lexical diversity, and β is another user-defined hyperparameter.

We experimented with penalizing 1-, 2-, 3-, and 4-grams equally but found it produced disfluent output, as the algorithm tended to avoid all words in the input. The exponential weight allows us to penalize the decoder for producing larger overlapping n-grams more harshly than small ones. All experiments in this work use $\beta = 4$, as this produced output in English which appeared fluent to the authors. Finally, the NMT model's vocabulary contains case variants (e.g., "his" and "His") and we do not want to add variation by trivially changing the case of words, so we penalize all case variants of the next tokens. Pseudocode for our approach is provided in Algorithm 1. Note that this method is much simpler than the method used to generate training data for ParaBank 2, which including hand-written constraints, scoring, filtering, and clustering.

[2] In this work, we assume words are separated by whitespace. For languages which do not denote word boundaries, our method could likely be applied after tokenizing the input, or by simply treating each SentencePiece token as a word.

[3] We apply the penalty at the start of the generation of the last word of an input n-gram so that the decoder is not encouraged to produce an unnatural completion to an already-begun word.

3.3 Diversity Control

The α parameter in Equation 1 provides the user with a knob to control how strongly the output is "pushed" away from the input, in lexical space, during generation. In contrast to positive and negative hard lexical lexical constraints (Hokamp and Liu, 2017; Post and Vilar, 2018; Hu et al., 2019c), our method requires no user-defined constraints, making it simpler and perhaps more language agnostic.[4]

3.4 Development and Evaluation

Paraphrase evaluation is complicated by the fact that many different aspects of paraphrases can be evaluated including semantic similarity between input and output, fluency, grammatical correctness, lexical diversity between input and output, and syntactic diversity between input and output. The relative importance of these aspects is not intuitively obvious and is likely determined by downstream tasks.

Modeling semantic similarity and lexical/syntactic diversity separately has the potential to somewhat lessen the burden of evaluation in several ways:

1. There are several potential ways to automatically evaluate the model $p(y \mid \mathcal{M}(x))$. One option is to evaluate perplexity on a test set consisting of human paraphrases. (Thompson and Post (2020) found that their multilingual NMT model assigned higher probability to both copies of the input and human paraphrases of the input, compared to a model trained on ParaBank 2.) Another option is to test models of $p(y \mid \mathcal{M}(x))$ on pairs of paraphrases where one paraphrase has been deemed to better preserve the semantic meaning of the input. Such datasets already exist, in about a dozen languages, due to the annotation efforts undertaken at the annual WMT evaluations.[5] In other words, we can simply treat a model of $p(y \mid \mathcal{M}(x))$ as an MT metric in order to judge its quality. In other words, we can simply treat a model of $p(y \mid \mathcal{M}(x))$ as an MT metric in order to judge its quality.
2. By applying the lexical/syntactic bias in generation, development of the generation algorithm can be conducted without the time/cost of re-training a model, and multiple generation schemes can be directly compared using the same $p(y \mid \mathcal{M}(x))$ model, such as the freely available Prism model (Thompson and Post, 2020).
3. Being able to control the amount of lexical and/or syntactic diversity at inference time allows for easier comparison with prior paraphrasing work, as the diversity can be adjusted to match that of a prior method. (We employ this approach in §4.3.1.)

4 Experimental Setup

4.1 Primary Model

We use the multilingual NMT model released with Prism (Thompson and Post, 2020), which uses a Transformer (Vaswani et al., 2017) architecture with approximately 750 million parameters. The model was trained in fairseq (Ott et al., 2019). The authors take several steps to encourage the encoder and decoder to be language agnostic, including specifying the target language as the first token in the target, so that the encoder does not know the target language, and training on several datasets that include a large number of different language pairs. The model was trained on several open source datasets including WikiMatrix (Schwenk et al., 2019), Global Voices,[6] EuroParl (Koehn, 2005) SETimes,[7] and United Nations. After filtering, this resulted in approximately 100 million translation pairs and covering 39 languages. The model uses a shared, multilingual vocabulary of 64k SentencePiece tokens (Kudo and Richardson, 2018).

4.2 Baseline Model

As a baseline, we train an English-only paraphraser in fairseq on the ParaBank 2 dataset (Hu et al., 2019c) with approximately 253M parameters and a SentencePiece vocabulary of 16k tokens. We train a Transformer with an 8-layer encoder, 8-layer decoder, 1024 dimensional embeddings, embedding sizes of 1024, feed-forward size of 4096, and 16 attention heads. Dropout is set to 0.3, label smooth-

[4] One concern with hard constraints is that there are sometimes words or phrases (e.g., proper nouns) that should not be paraphrased, as doing so would change the meaning of the sentence. Thus heuristics are often used to determine which words/phrases should be constrained.

[5] In particular, the relative ranking judgements collected through 2016 (Bojar et al., 2016) are probably the most relevant.

[6] http://casmacat.eu/corpus/global-voices.html

[7] http://nlp.ffzg.hr/resources/corpora/setimes/

Reference	Among other things, the developments in terms of turnover, employment, warehousing and prices are recorded.
α=0.0005	Among other things, developments in terms of turnover, employment, storage and prices are recorded.
α=0.003	Among other things, it records developments in turnover, employment, storage and prices.
α=0.006	Amongst other things, developments regarding turnover, employment, storage and prices were recorded.

Figure 1: Example English paraphrase for the three α values used in this work.

ing to 0.1, and learning rate to 0.0005, and batch size was 31200 tokens. Other parameters match the fairseq defaults. The model trained for approximately 6 weeks (33 epochs) on 4 Nvidia 2080 GPUs.

4.3 Evaluation

We conduct a manual evaluation in English using Mechanical Turk workers and conduct smaller scale manual evaluations in German and Spanish, with the help of colleagues who are native speakers. We perform human evaluations following (Hu et al., 2019b), described in more detail below.

4.3.1 English Evaluation

In this work, we focus on evaluation of semantic similarity, grammatical correctness, and lexical diversity. For the model trained on ParaBank 2, the trade-off between these dimensions is fixed and built into the model. To make a fair comparison, we adjust our overlap penalty (α) such that the output of our method matches the lexical diversity of the model trained on ParaBank 2. Following Hu et al. (2019c), we use uncased BLEU (Papineni et al., 2002), computed between input and output, to estimate the lexical diversity of the paraphraser.

We evaluate in English using Mechanical Turk workers who were selected from a curated list of previously vetted workers. Annotators were presented with a reference sentence and four paraphrases: three paraphrases from our proposed method (at three different operating points) and one from the model trained on ParaBank 2, presented in random order. For each paraphrase, the annotators were asked to (1) rate the paraphrase as (i) grammatical, (ii) having one or two small grammatical errors, or (iii) ungrammatical, and (2) rate the semantic similarity between the input and the paraphrase using an analog slider bar from 1–100. We randomly select 200 sentences from the English side of the WMT19 German–English test set (Barrault et al., 2019) and obtain ratings from three annotators, for each sentence at each paraphrase system/setting combination. Annotators were paid 0.50 USD per HIT.

For our proposed method, we choose three operating points: $\alpha = 0.0005$, α=0.003, and α=0.006 (Figure 1). The middle point of α=0.003 was chosen so as to produce output with the same lexical diversity as the paraphraser trained on ParaBank 2, as described above. We decode with a beam size of 5, using the fairseq defaults.

4.3.2 German & Spanish Evaluation

We also collect human judgments in German and Spanish. We follow the evaluation procedure described above for the English paraphraser except that annotations were done by colleagues who were native speakers in these languages. For Spanish, we used the target side of the WMT 2013 English–Spanish test set (Bojar et al., 2013). For German, we used the target side of the WMT 2019 English–German test set (Barrault et al., 2019). We obtained 50 judgments per set of 3 paraphrases by one German annotator, and 150 judgments per set of 3 paraphrases by three Spanish annotators, both on a random sample of sentences. Multiple paraphrases from our proposed method at different operating points (i.e., different values of α) were shown to the annotator, in random order.

5 Results

5.1 English Results

Human evaluation results in English are shown in Figure 2. We find that α is negatively correlated with grammaticality and semantic similarity between the input and output and positively correlated with lexical diversity of the output with respect to the input, as expected.

We find that at the operating point $\alpha = 0.003$, which was chosen such that our method has the same lexical diversity as the model trained on ParaBank 2, the paraphrases from our method were judged to be both more semantically similar to the input and grammatical (slightly) more often.

5.2 German & Spanish Results

The human evaluation results in German and Spanish, along with English for reference, are shown

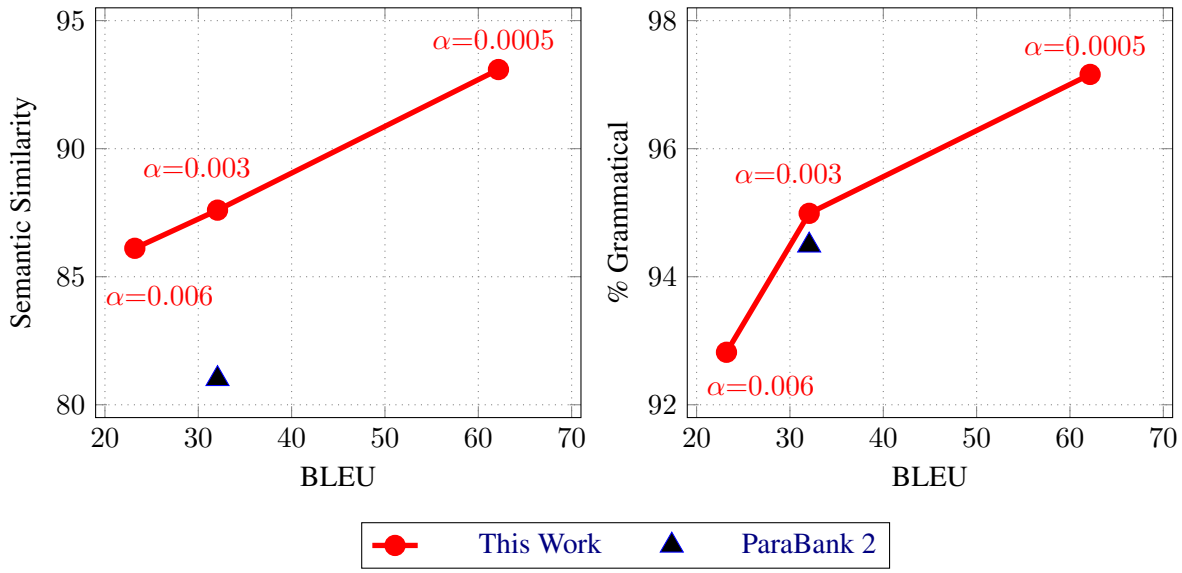


Figure 2: Human judgments of English paraphrases for semantic similarity (rated 1–100) and the percentage of sentences produced which were rated as grammatical, both as a function of lexical/syntactic diversity (measured via uncased BLEU between input and output). We evaluated our generation method at three operating points (α=0.0005, α=0.003, and α=0.006). α=0.003 was chosen to match such that the proposed method had the same diversity as the model trained on Paracrawl2. At that operating point, humans rated output of our method to be more semantically similar to the reference (87.5 vs. 81.0), and grammatical slightly more often (95.0% vs. 94.5%).

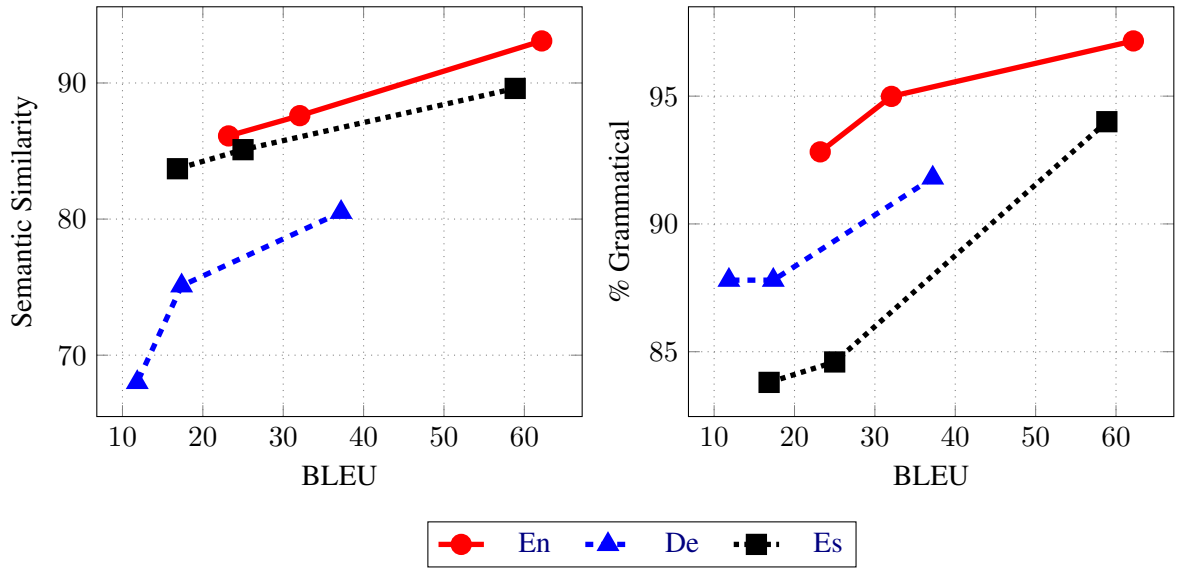


Figure 3: Human judgments of German (De) and Spanish (Es) paraphrases, with English (En) shown for reference, plotted against uncased BLEU computed between the paraphraser input and output. The judgement criteria and α values match English settings. α decreases from left to right in all plots.

in Figure 3. Note that we have no way to normalize between annotators in different languages, thus the results should *not* be used to draw conclusions about the *relative* performance of the paraphraser of these languages. However, we find the trends are similar across all three languages, and that semantic similarity and grammaticality judgements for Spanish and German are both reasonably high.

6 Discussion

We hypothesize that our method outperforms the baseline because it does not suffer from a fundamental shortcoming in creating synthetic paraphrase data from bitext: namely that inherent ambiguities present in one language (but not the other) can cause erroneous synthetic paraphrases in the

other language (Aziz and Specia, 2013).

For the sake of discussion we consider gender[8] as an ambiguity. Suppose we create synthetic English paraphrases from Turkish–English data, and our bitext contains the following (valid) sentence pair: ("O mağazaya gitti.", "She went to the store.") Turkish is a gender-neutral language, so when we translate the Turkish side to English it is perfectly valid to translate the sentence to "He went to the store." Pairing the original English translation with the translation results in the synthetic paraphrase example ("She went to the store.", "He went to the store."). Since English is gendered, this results in an invalid synthetic paraphrase.

In contrast, consider what happens if "She went to the store." is paraphrased by our method. First, the sentence is converted to an intermediate representation by the encoder. If the encoder were from an English→Turkish system, it is plausible that the encoder would discard gender information, as it is not needed in the target language. However, our encoder comes from a multilingual system which can produce output in *many* different languages. Thus, as long as the model has seen a sufficient number of training examples between English and at least one other gendered language, we can reasonably expect that the intermediate representation will preserve gender. Thus, when this representation is passed to the decoder and English is requested as the target language, the model should put low probability on any output for which the subject is male.

An alternative way to address pivot language ambiguities is to use multiple pivot languages, as proposed by Aziz and Specia (2013). However, it is not clear how best to extend this idea to neural sequence-to-sequence models, or to a multilingual paraphraser. Combining synthetic paraphrases for training using several different pivot languages would mitigate the errors due to ambiguities from any one pivot language, at the expense of errors due to ambiguities in other pivot languages. To really address such errors would require combining models of different language pairs; see Mallinson et al. (2017) for one such solution.

7 Conclusions

We treat paraphrasing as a zero-shot translation task and present a method to control the lexical diversity of paraphrases generated from a multilingual NMT model, enabling paraphrase generation in many languages. Our approach gives a user fine-grained control over the amount of lexical diversity at generation time, and also allows models and generation algorithms to be developed and evaluated with less interdependencies. There are likely many other ways that the output could be controlled to vary other aspects, such as syntactic diversity (Shu et al., 2019); we would like to explore such methods in future work.

Our work outperforms an English baseline trained on a large synthetic paraphrase dataset (Hu et al., 2019b). This improvement in performance may be because our method does not suffer from the issue that ambiguities in the pivot language used to create synthetic paraphrase data can cause errors in synthetic data. Small experiments indicate our method also performs well in other languages.

Multilingual NMT is an active research area and we are optimistic that this approach will pave the way for even stronger paraphrase generation in the future, as multilingual NMT methods continue to improve and models are publicly released.

Acknowledgments

The authors wish to thank Ben Van Durme for helpful technical discussions, and Carlos Aguirre, Sabrina Mielke, Rachel Wicks, and others for assistance with annotations.

[8] Czech is, of course, gendered, so we would not expect the ParaBank 2 dataset (which was created from Czech–English bitext) to have gender errors. But the logic presented here should generalize to other ambiguities.

References

Wilker Aziz and Lucia Specia. 2013. Multilingual WSD-like constraints for paraphrase extraction. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 202–211, Sofia, Bulgaria. Association for Computational Linguistics.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Loïc Barrault, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. 2019. Findings of the 2019 conference on machine translation (WMT19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 1–61, Florence, Italy. Association for Computational Linguistics.

Rahul Bhagat and Eduard Hovy. 2013. Squibs: What is a paraphrase? *Computational Linguistics*, 39(3):463–472.

Ondřej Bojar, Christian Buck, Chris Callison-Burch, Christian Federmann, Barry Haddow, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. 2013. Findings of the 2013 Workshop on Statistical Machine Translation. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 1–44, Sofia, Bulgaria. Association for Computational Linguistics.

Ondřej Bojar, Yvette Graham, Amir Kamran, and Miloš Stanojević. 2016. Results of the WMT16 metrics shared task. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 199–231, Berlin, Germany. Association for Computational Linguistics.

Mingda Chen, Qingming Tang, Sam Wiseman, and Kevin Gimpel. 2019a. Controllable paraphrase generation with a syntactic exemplar. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5972–5984, Florence, Italy. Association for Computational Linguistics.

Mingda Chen, Qingming Tang, Sam Wiseman, and Kevin Gimpel. 2019b. A multi-task approach for disentangling syntax and semantics in sentence representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2453–2464, Minneapolis, Minnesota. Association for Computational Linguistics.

Michael Denkowski and Alon Lavie. 2010. Extending the METEOR machine translation evaluation metric to the phrase level. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 250–253, Los Angeles, California. Association for Computational Linguistics.

Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. 2015. Multi-task learning for multiple language translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1723–1732, Beijing, China. Association for Computational Linguistics.

Li Dong, Jonathan Mallinson, Siva Reddy, and Mirella Lapata. 2017. Learning to paraphrase for question answering. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 875–886, Copenhagen, Denmark. Association for Computational Linguistics.

Wee Chung Gan and Hwee Tou Ng. 2019. Improving the robustness of question answering systems to question paraphrasing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6065–6075, Florence, Italy. Association for Computational Linguistics.

Jiatao Gu, Hany Hassan, Jacob Devlin, and Victor O.K. Li. 2018. Universal neural machine translation for extremely low resource languages. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 344–354, New Orleans, Louisiana. Association for Computational Linguistics.

Ankush Gupta, Arvind Agarwal, Prawaan Singh, and Piyush Rai. 2018. A deep generative framework for paraphrase generation. In *AAAI Conference on Artificial Intelligence*.

Chris Hokamp and Qun Liu. 2017. Lexically constrained decoding for sequence generation using grid beam search. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1535–1546, Vancouver, Canada. Association for Computational Linguistics.

J. Edward Hu, Huda Khayrallah, Ryan Culkin, Patrick Xia, Tongfei Chen, Matt Post, and Benjamin Van Durme. 2019a. Improved lexically constrained decoding for translation and monolingual rewriting. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 839–850, Minneapolis, Minnesota. Association for Computational Linguistics.

J. Edward Hu, Rachel Rudinger, Matt Post, and Benjamin Van Durme. 2019b. ParaBank: Monolingual bitext generation and sentential paraphrasing via lexically-constrained neural machine translation. In *Proceedings of AAAI*.

J. Edward Hu, Abhinav Singh, Nils Holzenberger, Matt Post, and Benjamin Van Durme. 2019c. Large-scale, diverse, paraphrastic bitexts via sampling and clustering. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 44–54, Hong Kong, China. Association for Computational Linguistics.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. 2018. Adversarial example generation with syntactically controlled paraphrase networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1875–1885, New Orleans, Louisiana. Association for Computational Linguistics.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Tomoyuki Kajiwara. 2019. Negative lexically constrained decoding for paraphrase generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6047–6052, Florence, Italy. Association for Computational Linguistics.

Huda Khayrallah, Brian Thompson, Matt Post, and Philipp Koehn. 2020. Simulated multiple reference training improves low-resource machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, Online. Association for Computational Linguistics.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86. Citeseer.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Zichao Li, Xin Jiang, Lifeng Shang, and Hang Li. 2018. Paraphrase generation with deep reinforcement learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3865–3878, Brussels, Belgium. Association for Computational Linguistics.

Jonathan Mallinson, Rico Sennrich, and Mirella Lapata. 2017. Paraphrasing revisited with neural machine translation. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 881–893, Valencia, Spain. Association for Computational Linguistics.

Kathleen R. McKeown. 1983. Paraphrasing questions using given and new information. *American Journal of Computational Linguistics*, 9(1):1–10.

Tong Niu and Mohit Bansal. 2018. Adversarial over-sensitivity and over-stability strategies for dialogue models. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 486–496, Brussels, Belgium. Association for Computational Linguistics.

Tong Niu and Mohit Bansal. 2019. Automatically learning data augmentation policies for dialogue tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1317–1323, Hong Kong, China. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Ngoc-Quan Pham, Jan Niehues, Thanh-Le Ha, and Alexander Waibel. 2019. Improving zero-shot translation with language-independent constraints. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 13–23, Florence, Italy. Association for Computational Linguistics.

Matt Post and David Vilar. 2018. Fast lexically constrained decoding with dynamic beam allocation for neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1314–1324, New Orleans, Louisiana. Association for Computational Linguistics.

Chris Quirk, Chris Brockett, and William Dolan. 2004. Monolingual machine translation for paraphrase generation. In *Proceedings of the 2004 Conference*

on Empirical Methods in Natural Language Processing, pages 142–149, Barcelona, Spain. Association for Computational Linguistics.

Alessandro Raganato, Raúl Vázquez, Mathias Creutz, and Jörg Tiedemann. 2019. An evaluation of language-agnostic inner-attention-based representations in machine translation. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 27–32, Florence, Italy. Association for Computational Linguistics.

Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán. 2019. Wikimatrix: Mining 135m parallel sentences in 1620 language pairs from wikipedia. *CoRR*, abs/1907.05791.

Holger Schwenk and Matthijs Douze. 2017. Learning joint multilingual sentence representations with neural machine translation. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 157–167, Vancouver, Canada. Association for Computational Linguistics.

Raphael Shu, Hideki Nakayama, and Kyunghyun Cho. 2019. Generating diverse translations with sentence codes. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1823–1827, Florence, Italy. Association for Computational Linguistics.

Brian Thompson and Matt Post. 2020. Automatic machine translation evaluation in many languages via zero-shot paraphrasing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, Online. Association for Computational Linguistics.

Jörg Tiedemann and Yves Scherrer. 2019. Measuring semantic abstraction of multilingual NMT with paraphrase recognition and generation tasks. In *Proceedings of the 3rd Workshop on Evaluating Vector Space Representations for NLP*, pages 35–42, Minneapolis, USA. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

John Wieting and Kevin Gimpel. 2018. ParaNMT-50M: Pushing the limits of paraphrastic sentence embeddings with millions of machine translations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 451–462, Melbourne, Australia. Association for Computational Linguistics.

John Wieting, Kevin Gimpel, Graham Neubig, and Taylor Berg-Kirkpatrick. 2019. Simple and effective paraphrastic similarity from parallel translations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4602–4608, Florence, Italy. Association for Computational Linguistics.

John Wieting, Jonathan Mallinson, and Kevin Gimpel. 2017. Learning paraphrastic sentence embeddings from back-translated bitext. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 274–285, Copenhagen, Denmark. Association for Computational Linguistics.

Liang Zhou, Chin-Yew Lin, Dragos Stefan Munteanu, and Eduard Hovy. 2006. ParaEval: Using paraphrases to evaluate summaries automatically. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 447–454, New York City, USA. Association for Computational Linguistics.

Zhong Zhou, Matthias Sperber, and Alexander Waibel. 2019. Paraphrases as foreign languages in multilingual neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 113–122, Florence, Italy. Association for Computational Linguistics.

When Does Unsupervised Machine Translation Work?

Kelly Marchisio and **Kevin Duh** and **Philipp Koehn**
Johns Hopkins University
kmarc@jhu.edu, kevinduh@cs.jhu.edu, phi@jhu.edu

Abstract

Despite the reported success of unsupervised machine translation, the field has yet to examine the conditions under which the methods succeed and fail. We conduct an extensive empirical evaluation using dissimilar language pairs, dissimilar domains, and diverse datasets. We find that performance rapidly deteriorates when source and target corpora are from different domains, and that stochasticity during embedding training can dramatically affect downstream results. We advocate for extensive empirical evaluation of unsupervised MT systems to highlight failure points and encourage continued research on the most promising paradigms. Towards this goal, we release our preprocessed dataset to stress-test systems under multiple data conditions.

1 Introduction

Machine translation (MT) has progressed rapidly since the advent of neural machine translation (NMT) (Kalchbrenner and Blunsom, 2013; Bahdanau et al., 2015; Sutskever et al., 2014) and is better than ever for languages for which ample high-quality bitext exists. Conversely, MT for low-resource languages remains a great challenge due to a dearth of parallel training corpora and poor quality bitext from esoteric domains. To address this, several authors have proposed unsupervised MT techniques, which rely only on monolingual text for training (e.g., Ravi and Knight, 2011; Yang et al., 2018; Artetxe et al., 2018c; Hoshen and Wolf, 2018; Lample et al., 2018a,b; Artetxe et al., 2018b, 2019).

Recent unsupervised MT results appear promising, but they primarily report results for the high-resource languages for which traditional MT already works well. The limits of these methods are so far under-explored. For unsupervised MT to be a viable path for low-resource machine translation, the field must determine (1) if it works outside highly-controlled environments, and (2) how to effectively evaluate newly-proposed training paradigms to pursue those which are promising for real-world low-resource scenarios. Unsupervised MT methods must work (1) on **different scripts** and between **dissimilar languages**, (2) with **imperfect domain alignment** between source and target corpora, (3) with a **domain mismatch** between training data and the test set, and (4) on the low-quality data of **real low-resource languages**. These factors reflect the real-life challenges of low-resource translation.

Our main contribution is an extensive analysis of unsupervised MT with regards to factors (1)-(3) above.[1] We find that (a) translation performance rapidly deteriorates when source and target corpora are from different domains, (b) stochasticity during word embedding training can dramatically affect downstream bilingual lexicon induction (BLI) and translation performance, and (c) like in the bilingual lexicon induction literature, unsupervised MT performance declines when source and target languages are dissimilar. While (4) is not the focus of this paper, we do observe very low performance on an authentic low-resource language pair, corroborating previous studies (Guzmán et al., 2019).

Finally, as there are no standard evaluation protocols to ensure that unsupervised MT systems are robust to the types of data anomalies ubiquitous in low-resource translation settings, we advocate for extensive empirical evaluation of unsupervised MT systems to highlight failure points and encourage continued research on the most promising paradigms.

We first discuss related work in Section 2, followed by a detailed overview of the unsupervised MT architecture in Section 3. In Section 4, we discuss our research questions, followed by our evaluation methodology and datasets in Sections 5

[1] We release our full dataset at http://statmt.org/when-does-unsup-work to facilitate the stress-testing of systems.

Proceedings of the 5th Conference on Machine Translation (WMT), pages 571–583
Online, November 19–20, 2020.

and 6. Section 7 presents our findings, and Section 8 discusses the results. We conclude in Section 9.

2 Related Work

Bilingual Lexicon Induction Unsupervised MT methods can be thought of as an end-to-end extension of work inducing bilingual lexicons from monolingual corpora. Bilingual lexicon induction (BLI) using non-parallel data has a rich history, beginning with corpus statistic and decipherment methods (e.g., Rapp, 1995; Fung, 1995; Koehn and Knight, 2000, 2002; Haghighi et al., 2008), continuing to modern neural methods to create crosslingual word embeddings (e.g. Mikolov et al., 2013a; Conneau et al., 2018, see Ruder et al. (2019) for a survey) which form a critical component of state-of-the-art unsupervised MT systems.

Evaluation of Embedding Spaces Søgaard et al. (2018) determine that monolingual embedding spaces of similar languages are not typically isomorphic as was previously believed, and that bilingual dictionary induction "depends heavily on... the language pair, the comparability of the monolingual corpora, and the parameters of the word embedding algorithms." Vulić et al. (2019) argue that unsupervised approaches are unsuccessful with dissimilar languages and domains, and that unsupervised performance has been overly lauded because the conditions under which they were compared with supervised baselines were inequitable.

While a modest body of literature has examined the quality of cross-lingual word embeddings (CLEs) by measuring performance on BLI, Glavaš et al. (2019) evaluate on downstream natural language tasks, underlining the importance of full-system evaluation. The authors conclude that "the quality of CLE models is largely task-dependent and that overfitting the models to the BLI task can result in deteriorated performance in downstream tasks." Similarly, Doval et al. (2019) investigate cross-lingual natural language inference.

Evaluation of Unsupervised MT Liu et al. (2020) helpfully re-define unsupervised machine translation into three distinct categories: (1) no bitext whatsoever, (2) the target language pair is linked through bitext via a pivot language, and (3) no linkage through a pivot language, but bitexts exists for *some* language and the target language. The authors analyze their multilingual pretraining method with respect to other similar training paradigms (Conneau and Lample, 2019; Song et al., 2019) and evaluate unsupervised MT performance when using backtranslation (Definition 1) or language transfer after finetuning on related bitext (Definition 3).

In unsupervised MT with no bitext, Lample et al. (2018b) ablate their PBSMT system, finding that initial phrase table quality is critical and that performance suffers when the language model is trained with less data. They tweak their NMT embedding initialization method, such as using separately-trained BPE instead of joint, and word embeddings instead of BPE token embeddings. They report the results of dropping part of their loss function and making minor changes to the NMT architecture on downstream BLEU score. Concurrently to our work, Kim et al. (2020) arrived at similar conclusions to us using a different autoencoder/dual-learning unsupervised MT approach based on cross-lingual language model pretraining (Conneau and Lample, 2019); this complements our experiments and corroborates our results.

3 Background: Unsupervised MT

Our experiments employ the models of Artetxe et al. (2018b, 2019) as representative of state-of-the-art for the class of unsupervised MT methods that bootstrap from cross-lingual word embeddings. Recent work such as Lample et al. (2018b) is based on similar concepts. For our purposes, unsupervised MT follows Liu et al. (2020)'s Definition (1) from Section 2, where no bitext exists.

Another approach to unsupervised MT involves pretraining a bilingual or multilingual model on monolingual text on a general task before finetuning on translation. Such methods include cross-lingual language model pretraining (Conneau and Lample, 2019), masked sequence-to-sequence pretraining (Song et al., 2019), and multilingual denoising pretraining (Liu et al., 2020), and have shown promise. For instance, Liu et al. (2020) record the first good results on the low-resource Sinhala-English and Nepali-English pairs. While pretraining and multilingual methods are not the subject of this work, they warrant future evaluation.

Figure 1 depicts the basic training process. It is the publicly-available SMT setup of Artetxe et al. (2018b)[2], plus the "NMT hybridization" steps from Artetxe et al. (2019).[3]

[2] https://github.com/artetxem/monoses

[3] Shared with us by Mikel Artetxe.

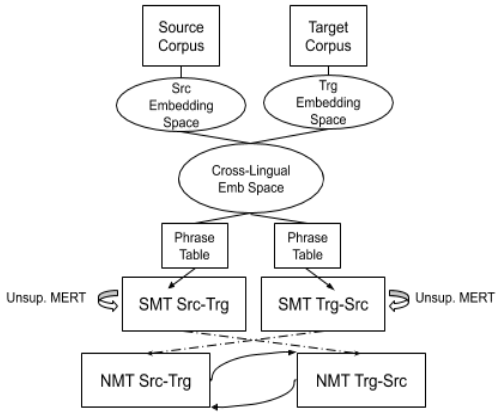


Figure 1: The unsupervised MT architecture used in this work. This model is a replication of Artetxe et al. (2018b) [steps before NMT] and Artetxe et al. (2019) [NMT component].

Training begins with two monolingual corpora which are not necessarily related in any way (i.e. they are not assumed to be parallel nor comparable text). First, word embeddings are trained independently for each corpus, resulting in a source and a target embedding space. Specifically, after preprocessing, Artetxe et al. (2018b) train two statistical language models using KenLM (Heafield, 2011), one for the source language and one for the target. They use phrase2vec[4] (Artetxe et al., 2018b), an extension of Mikolov et al. (2013b)'s skip-gram model,[5] to generate phrase embeddings for 200,000 unigrams, 400,000 bigrams, and 400,000 trigrams.

Next, source and target word embeddings are aligned into a common cross-lingual embedding space. They run VecMap[6] (Artetxe et al., 2018a) which calculates a linear mapping of one space to another based on the intuition that phrases with similar meaning should have similar neighbors regardless of language. Given a matrix of source word embeddings X and target word embeddings Z which have been length-normalized, mean-centered, then length-normalized again, VecMap calculates $M_x = XX^T$ and $M_z = ZZ^T$. Each cell $M_{x_{ij}}$ and $M_{z_{ij}}$ is the cosine similarity between words X_i and X_j, and Z_i and Z_j, respectively. M_x and M_z are symmetric, and if the monolingual vector spaces were fully isometric, M_x and M_z would be identical besides rows and columns being permuted. Each row of M_x and M_z is a similarity distribution. To exploit this, each row of $\sqrt{M_x}$ and $\sqrt{M_z}$ is sorted (they find that using the square root works better empirically), and length-normalized, mean-centered, and length-normalized again. For each row i in $sorted(\sqrt{M_x})$, they find the row j of $sorted(\sqrt{M_z})$ that is its nearest neighbor, and assign $X_i = Z_j$ in the initial translation dictionary D. A cell $D_{ij} = 1$ if words X_i, and Z_j are translations of one another, and 0 otherwise.

Next, there is an iterative process of calculating the optimal linear mappings and extracting an updated dictionary. For calculating the mapping, the goal is to find the linear transformations W_x and W_z which maximize the cosine similarity of the words that are translations of one another as defined by the dictionary D, over the entire dictionary:

$$\arg\max_{W_x,W_z} \sum_i \sum_j (D_{ij})((X_i W_x) \cdot (Z_j W_z))$$

From there, they calculate $M = XW_x W_z^T Z^T$, whereby each cell in M is the cosine similarity of word X_i and Z_j after their transformations with W_x and W_z. To avoid poor local optima, they stochastically zero-out some cells of M with probability $p = 0.9$, decreasing over time.

The final score for each potential translation candidate is calculated using Cross-domain Similarity Local Scaling (CSLS) (Conneau et al., 2018) to mitigate the hubness problem. CSLS utilizes cosine similarity, which is taken from M. For each pair of words X_i and Z_j, the new dictionary cell $D_{ij} = 1$ if the CSLS score between X_i and Z_j is the highest over all other words in Z, and $D_{ij} = 0$ otherwise. The dictionary is created in both directions, and concatenated. Readers are directed to Artetxe et al. (2018a) for further details.

The next step extracts an initial phrase-table for use in a SMT system. They use the softmax over the cosine similarity of the 100 nearest-neighbors of each source phrase embedding as the phrase translation probabilities. This is done in both directions:

$$(f|e) = \frac{e^{(\cos(e,f)/\tau)}}{\sum_{f'} e^{(\cos(e,f')/\tau)}}$$

For the target embedding with the highest cosine similarity, the phrases are aligned, and unigram translation probabilities are multiplied to become the lexical weighting.

Combining the preliminary phrase table with a distortion penalty and language model produces the initial unsupervised phrase-based SMT system (Koehn et al., 2007). The SMT model weights

[4] https://github.com/artetxem/phrase2vec

[5] https://github.com/tmikolov/word2vec

[6] https://github.com/artetxem/vecmap

are tuned using a variant of MERT (Och, 2003) designed for unsupervised scenarios, which uses 10,000 parallel sentences generated via backtranslation (Sennrich et al., 2016a). The SMT model then undergoes three rounds of iterative backtranslation.

Artetxe et al. (2019) extend their 2018 work by adding a critical "NMT hybridization" final step, which achieves significant gains over SMT alone.[7] An NMT system is trained using backtranslated output from SMT for one epoch. On the next epoch, a small number of sentences are backtranslated with the newly-trained NMT system and concatenated with a slightly smaller fraction of SMT-generated bitext. The procedure continues for 30 epochs, gradually increasing the percentage of synthetic training data created by the NMT system until all of the training data is NMT-generated. The NMT system is trained for an additional 30 epochs of iterative backtranslation using data generated fully by the NMT system of the previous epoch. The test set is translated with beam search using an ensemble of models saved at every tenth epoch (six total), resulting in BLEU scores of 33.2 and 26.4 (SacreBLEU (Post, 2018)) on newstest2014 for French-English and German-English, respectively.

We run Artetxe et al. (2018b, 2019)'s implementation for our experiments. Specifically, neural models are Transformer-big (Vaswani et al., 2017) trained with fairseq (Ott et al., 2019) on one NVIDIA GeForce GTX 1080Ti GPU. Models use shared embeddings, the Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.98$ (Kingma and Ba, 2015), label smoothing, initial learning rate of 1e-07 warming up for 4000 steps to 5e-04 before decaying, and dropout (Srivastava et al., 2014) probability of 0.3. We set optimizer delay to 4 to simulate 4 GPUs.

To elucidate the performance gap due to the unsupervised architecture, we build a standard supervised NMT system using the same neural architecture described above. We train until performance on the development set ceases to improve for 10 epochs. To parallel the unsupervised setup, we translate the test set using an ensemble of 6 models; We perform ensemble selection by performance on a validation set, selecting the best-performing checkpoint along with 5 previous checkpoints.

[7] Readers are directed to Artetxe et al. (2019) for additional changes that resulted in sizable BLEU (Papineni et al., 2002) gains before the NMT phase.

4 Research Questions

Existing unsupervised translation methods work well on languages which are similar to each other, use the same Roman script, and have an ample amount of monolingual news data available (which matches the test set domain). Questions remain as to whether unsupervised methods will be useful on authentic low-resource settings where few or none of the aforementioned properties hold. Namely, does unsupervised MT work with:

- dissimilar languages?
- dissimilar source and target domains?
- diverse datasets?
- authentic low-resource language pairs?

Such questions reflect the reality of authentic low-resource translation, and are those which must be adequately resolved for unsupervised MT to be a viable alternative to traditional translation methods for the most difficult language pairs.

5 Evaluation of Unsupervised MT

We perform an extensive empirical evaluation of unsupervised MT. Our evaluation protocol stress-tests an unsupervised MT system under varying conditions to reveal its points of strength and failure. Systems should be judged on how well they perform: (1) on dissimilar languages, (2) on increasingly divergent domains between source and target corpora, (3) on diverse datasets, and (4) on authentic low-resource language pairs where data quality is typically low. Namely, we:

1. Choose 2 language pairs, at least one of which where the source and target languages utilize different scripts.
2. Choose 3 datasets of different domains, at least one of which is parallel bitext.
3. Perform at least one experiment for each language pair under each of the following data conditions:
   - Originally parallel
   - Not originally parallel
   - Different domain for source and target.
4. Choose 2 *true* low-resource language pairs.

5. Judge the system based on performance in all tested scenarios.

The data conditions above are designed measure how well a system performs in regards to the research questions of Section 4. Namely, success on a variety of languages with different scripts and linguistic structure indicates robustness to dissimilar languages; success on multiple datasets of different domains indicates that the system is not specifically designed for one domain at the expense of others, and performs well even when training and test data do not match perfectly; Step #3 evaluates performance on increasingly divergent domains between source and target data; and Step #4 is the *true* test—whether the system succeeds on authentic low-resource language pairs.

6 Datasets

Training datasets used in our reinvestigation of the unsupervised MT system presented in Artetxe et al. (2019) are shown in Table 1. We focus on Russian-English (Ru-En) and French-English (Fr-En) tasks and include as reference Sinhala-English (Si-En) and Nepali-English (Ne-En) as well. Following Section 5, we evaluate the same system under various ablated data setups:

- The "Supervised" condition is the standard MT training setup which uses parallel bitext.
- In the "Parallel" condition, an *unsupervised* MT system is trained on a corpus that was originally parallel (i.e. UN corpus), now being treated as two separate monolingual corpora.
- In contrast, the "Disjoint" setting splits data from a parallel corpus into two disjoint halves, using the first half of the source-side corpus and the second half of the target-side corpus.
- In the "Different Domain" (Diff. Dom.) setting, source and target monolingual corpora come from different domains. This is a realistic setting in low-resource scenarios, and is expected to be much more difficult than the "Disjoint" setting.
- "News crawl" (News) and "Common Crawl" (CC) settings determine whether the system can flexibly handle diverse datasets.

Specifics of the datasets used are described in subsequent subsections. Token counts presented in the subsections below are before preprocessing, whereas Table 1 reflects the data remaining after the preprocessing procedure of Artetxe et al. (2018b). We will release the preprocessed data splits for others to compare their results with ours.

Condition		Corpus	Src	Trg
Repro	Fr-En	News	694	1940
	En-Fr	News	1940	694
Supervised	Fr-En	UN: A	346	301
	Ru-En	UN: A	284	284
Parallel	Fr-En	UN: A	302	270
	Ru-En	UN: A	232	241
Disjoint	Fr-En	UN: A / B	302	255
	Ru-En	UN: A / B	232	236
Diff. Dom.	Fr-En	UN: A / CC	302	226
	Ru-En	UN: A / CC	232	226
News	Fr-En	News	116	105
	Ru-En	News	120	105
CC	Fr-En	CC	110	79
	Ru-En	CC	115	79

Table 1: Training data after preprocessing. UN = United Nations, CC = Common Crawl, News = News crawl. "Diff. Dom." uses UN on the source-side and CC on the target-side. "News" is a subset of 2007-08 for En, 2007-09 for Fr, and 2008-11 for Ru. "Repro" is the condition most similar to (Artetxe et al., 2018b, 2019). Src (M) and Trg (M) columns are the token counts, in millions. "Supervised" count is in BPE tokens. All others are token counts for SMT (pre-BPE).

6.1 United Nations

The United Nations Parallel Corpus (UN) (Ziemski et al., 2016) contains official United Nations documents from 1990-2014, human-translated into six languages. The first 10,000 lines of each dataset are held-out. The remaining lines are partitioned into training sets A & B. Training set A on the source side and A on the target side are paired to form the Parallel training set; Training set A on the source side and B on the target side are paired to form the Disjoint training set.

6.2 News Crawl

News crawl (News) consists of monolingual data crawled from news websites. Data for each year has been shuffled. Following Artetxe et al. (2018b), we concatenate News crawl 2007-13 for English and for French. For Russian, we concatenate News crawl 2008-18. We use the deduplicated Russian corpus. We use the full datasets to reproduce

Artetxe et al. (2018b, 2019)'s work. For subsequent experiments, we use a subset: the first 100 million tokens from each concatenated News crawl corpus before preprocessing. For English, this is all of News crawl 2007 and ∼23.3 million tokens from News crawl 2008. For French, it is News crawl 2007, 2008, and some of 2009. For Russian, it is News crawl 2008-2010, and some of 2011.

6.3 Common Crawl

The Common Crawl (CC) corpora consists of web-scraped monolingual data ordered as documents. We extract two training datasets from the English corpus - one with the first ∼291 million tokens and another with the first ∼100 million for Diff. Dom and CC experiments, respectively. We do not shuffle this data, as having less documents better simulates real low-resource settings. Sinhala and Nepali contain approximately 103 million and 110 million tokens, as used in Guzmán et al. (2019). We additionally extract the first 100 million French and Russian tokens for CC experiments.

6.4 Preprocessing

Training data is preprocessed separately for each unsupervised experiment as part of Artetxe et al. (2018b)'s training pipeline. Data is deduplicated, and tokenized and truecased using scripts from Moses (Koehn et al., 2007). Sentences with less than 3 tokens or more than 80 tokens are discarded, and sentences are shuffled. Ten thousand sentences are removed to form a development set. To begin the NMT phase, a joint BPE (Sennrich et al., 2016b) vocabulary of 32000 tokens is learned. Source- and target-side corpora are backtranslated using the final model from the SMT phase, and all data then has BPE applied.[8]

For supervised experiments, training data is tokenized and truecased, and then a joint BPE (Sennrich et al., 2016b) vocabulary of 32000 tokens is learned. After applying BPE, the data is cleaned using Moses' `clean-corpus-n.perl`, discarding sentences under 3 and greater than 80 tokens.

6.5 Vocabulary Overlap of Training Sets

A vocabulary of unigrams was collected for each target-side (English) corpus, which includes tokens that appear at least 10 times, for a maximum of 200,000 unigrams. Of approximately 144,000 such unique tokens between UN-A and UN-B from the Fr-En UN corpus, the corpora share 54.1%. These corpora are used in the Disjoint condition. The respective vocabulary overlap for UN-A and CC from the Diff. Dom condition for Fr-En is 25.7%. For UN-B vs. CC for Fr-En, they share 25.3%. Statistics are analogous for Ru-En.

6.6 Test and Validation Sets

Ru-En models are tested on newstest2019. Fr-En models are tested on newstest2014. Supervised models use newstest2018 (Ru-En) or newstest2013 (Fr-En) for validation. For Si-En and Ne-En, we use the Wikipedia dev and devtest sets from Guzmán et al. (2019).[9] For supervised models, we select the ensemble with best performance on newstest2017 (Ru-En) or newstest2012 (Fr-En).

7 Reinvestigation of Artetxe et. al.

First, we replicate Artetxe et al. (2018b, 2019), achieving relatively comparable results (Table 2). Differences in BLEU scores are likely attributable to using Artetxe et al. (2018b)'s code for all steps before the NMT phase; Artetxe et al. (2019) improved upon these, but we chose to use the publicly available code from the previous year.

	Artetxe et al. (2019)	**This Work**
Fr-En	33.2	31.1
En-Fr	33.6	32.8

Table 2: Artetxe et al. (2019)'s unsupervised MT performance vs. the system in this work, which is a combination of Artetxe et al. (2018b) [steps before NMT] and Artetxe et al. (2019) [NMT component], using the full News crawl datasets from Subsection 6.2. Scored using SacreBLEU (Post, 2018) on newstest2014.

Next, we set up a series of experiments to assess the questions posed in Section 4. Results are presented in Tables 3 and 4.

7.1 Unsupervised Quality Loss

The Supervised ("Sup.") column of Table 3 shows performance of a standard Transformer-big architecture on parallel bitext for Ru-En and Fr-En. Assuming that supervised translation will always outperform unsupervised, these scores represent a ceiling to quantify how much potential quality is lost using an unsupervised architecture.

[8] Some experiments had Moses' `clean-corpus-n.perl` applied after this.

[9] https://github.com/facebookresearch/flores/raw/master/data/wikipedia_en_ne_si_test_sets.tgz

	Sup.	Parallel	Disjoint	Diff. Dom.
Corpus	A / A	A / A	A / B	A / CC*
Ru-En	26.9	23.7 *(-3.2)*	21.2 *(-5.7)*	0.7 *(-26.2)*
Fr-En	29.9	27.6 *(-2.3)*	27.0 *(-2.9)*	3.9 *(-26.0)*

Table 3: Unsupervised MT performance on a single run using the United Nations (UN) dataset. "Diff. Dom." uses UN data as source and Common Crawl (*) as target. "Sup." is supervised with UN parallel data. A / A refers to UN training dataset A used on the source and target sides, for example. Scored using SacreBLEU (Post, 2018) on newstest2019 (Ru-En) and newstest2014 (Fr-En).

.

The supervised models and those in the Parallel column use the same datasets[10] and can therefore be directly compared. We observe a BLEU score drop of ∼3.2 for Ru-En versus a drop of ∼2.3 for Fr-En when using the unsupervised architecture. This minor quality loss represents a strong result for unsupervised MT; however, the question is whether the results will remain strong as we gradually make the monolingual corpora less similar.

7.2 Investigating Our Research Questions

Does unsupervised machine translation work for:

(1) Dissimilar language pairs?

We conduct experiments in French and Russian into English. Whereas French and English share the same Roman script and common linguistic origin, Russian is a Slavic language that uses the Cyrillic script. The results presented in Tables 3 and 4 indicate that unsupervised MT is more difficult when writing script and language family differs. Across the board, we observe that the ΔBLEU between supervised and unsupervised performance is wider for Ru-En than for Fr-En, particularly for News and Common Crawl datasets. For instance, whereas Fr-En loses 2.9 BLEU in the Supervised versus Disjoint setups (which use comparable data), Ru-En loses 5.7 BLEU. While we acknowledge that in general one should not compare BLEU scores across language pairs or datasets, this gap suggests that unsupervised MT may behave differently for different language pairs.

(2) Dissimilar domains?

We investigate the effects of domain similarity between source and target training corpora. For each language, we observe the difference in performance on Table 3 of the Parallel, Disjoint, and Diff. Dom. columns.

Because training data in the Parallel condition was originally parallel, these experiments have the highest possible domain match between source and target data. Since Disjoint data was extracted from the same corpus but was not parallel, source and target can be thought of as having very slightly different domains. We observe a minor performance drop between Parallel and Disjoint experiments, which is more pronounced for Ru-En.

Examining the Diff. Dom. column, however, the performance contrast is stark. While both language pairs obtain respectable BLEU scores in the 20s when domains match in Parallel and Disjoint conditions, performance drops sharply when training set domains are mismatched—scoring 3.9 BLEU for Fr-En and 0.7 for Ru-En. (A subsequent run of Fr-En scored 17.4, addressed in Section 7.4). The fault is not with either side of the training corpus alone—Parallel/Disjoint experiments from Table 3 which use UN data alone and CC experiments in Table 4 which use Common Crawl data alone perform acceptably—it is when the two datasets are paired as source-target in Diff. Dom. conditions that performance rapidly deteriorates.

(3) Diverse datasets?

	UN	News	CC
Ru-En	21.2	16.1	13.8
Fr-En	27.0	28.2	22.4
Si-En	n/a	n/a	0.2
Ne-En	n/a	n/a	0.4

Table 4: Unsupervised MT performance on a single run using diverse datasets [UN = United Nations (Disjoint), News = News Crawl, CC = Common Crawl]. Scored using SacreBLEU (Post, 2018) on newstest2019 (Ru-En), newstest2014 (Fr-En), and the FLoRes Wikipedia evaluation sets (Si-En, Ne-En) (Guzmán et al., 2019).

Table 4 shows the results of experiments using three different training datasets. News crawl matches the domain of the test set exactly. UN data has a moderate domain match with the test set, and CC matches the least. Not unexpectedly, most experiments where training and test domain match perform better than when there is a domain mismatch. The exception is the News experiment for Ru-En, where the model performs considerably worse than the UN condition despite having a stronger domain match. Notably, News has approx-

[10] Differences in token count are due to the different preprocessing detailed in Section 6.4.

imately 2-3x less data than UN for each language pair. We suspect that for Fr-En, the relative ease of unsupervised translation for this language pair allowed the strong domain match with the test set to outweigh the lower amount of data. On the other hand, the relative difficulty of unsupervised MT in Ru-En made the system suffer too greatly in the lower-resource condition, to where it could not compensate with domain match.

(4) A true low-resource pair?

Facebook recently released test sets for Sinhala-English and Nepali-English, true low-resource language pairs which not only lack bitext, but monolingual data is of poor quality. These languages do not share a script or language family with English, and the data is out-of-domain with the English data. This reflects a real-world low-resource scenario where we would hope to benefit from unsupervised MT. We observe extremely poor results in Table 4, with Si-En achieving a BLEU score of 0.2, and 0.4 for Ne-En. Guzmán et al. (2019) achieve similarly poor results for these language pairs without using supplemental data from a related language.

7.3 BLEU During Training

Figure 2 shows translation performance for the experiments in Tables 3 and 4 at various steps during the unsupervised machine translation pipeline. Most SMT models improve performance slightly as a result of unsupervised MERT tuning, and more substantially after three rounds of iterative back-translation. Substantial improvement occurs as a result of NMT training for all models except the degenerate Diff. Dom conditions.

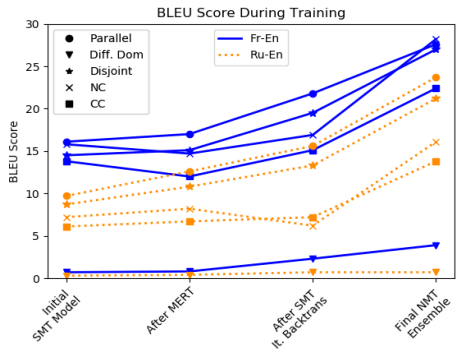


Figure 2: BLEU score during training.

7.4 Training Stability

One challenge with unsupervised methods is training stability: stochasticity during training can give substantially different results due to the iterative bootstrap nature of the training process.

	Condition	Min	Max	μ	σ
En-Fr	Repro	33.08	42.47	40.86	2.5
Fr-En	Repro	45.21	46.92	46.06	0.47
	Parallel	48.0	50.2	49.09	0.69
	Disjoint	37.88	39.09	38.47	0.37
	Diff. Dom.	**0.0**	**17.27**	**7.97**	**7.95**
	News	25.86	28.1	26.97	0.56
	CC	25.87	27.6	26.9	0.51
Ru-En	Parallel	32.24	34.04	32.95	0.47
	Disjoint	25.08	26.96	25.79	0.58
	Diff. Dom.	0.0	0.1	0.01	0.03
	News	22.19	23.77	23.1	0.44
	CC	**0.0**	**24.69**	**12.61**	**11.45**

Table 5: Accuracies (%) of induced dictionaries on 10-11 runs. Bold experiments were severely unstable.

In their analysis of unsupervised methods for generating CLEs, Glavaš et al. (2019) note considerable instability in performance on BLI. Defining failure as having a mean average precision (MAP) of $<$0.05 on all training runs, Iterative Closest Point (Hoshen and Wolf, 2018) fails for $\sim$21% of language pairs, Gromov-Wasserstein Alignment (Alvarez-Melis and Jaakkola, 2018) for $\sim$46%, and MUSE (Conneau et al., 2018) for $\sim$54%. VecMap (Artetxe et al., 2018a) succeeds for all language pairs, leading Glavaš et al. to deem it the most robust. Artetxe et al. (2018a) demonstrate their robustness over other methods in their work. When counting successful runs as achieving $>$5.0% accuracy, VecMap is successful 10/10 times for three language pairs. Hartmann et al. (2019) also investigate instability in vector space alignment methods.

After training phrase embeddings for each experiment, we run VecMap on the generated embedding spaces ten additional times and indeed find little fluctuation in BLI between runs. When rerunning the full pipeline for each experiment, however, we observe considerable instability in two experiments which dramatically affects downstream results.

We build a gold-standard bilingual dictionary of 2000 word pairs from Wikipedia data (Wołk and Marasek, 2014) available publicly on OPUS (Tiedemann, 2012), and run the first four steps of the unsupervised training procedure additional times for each experiment. Table 5 contains the summary results of 10-11 runs of each experiment.

Tables 3 and 4 present the results of the single

first run of each experiment. Whereas the majority have consistent accuracy on bilingual lexicon between runs as seen in Table 5, Diff. Dom. for Fr-En and CC for Ru-En are highly unstable. The BLI accuracy of additional runs of Fr-En Diff. Dom. ranged between 0.0% and 17.27%. Of the initial run and 9 subsequent, five had accuracies <0.1%, while the other five had accuracies >15.26%. For Ru-En CC, the run reported in Table 4 had a BLI accuracy of 21.35%. Of eleven runs, five had an accuracy <0.26%, and six had an accuracy >21.35%.

As evidence of the critical effect of BLI accuracy on downstream BLEU, whereas the Fr-En Diff. Dom. run reported in Table 3 had a BLI accuracy of 0.0%, a subsequent run of the entire training pipeline had an accuracy of 17.08% and a final BLEU score of 17.4. (This experiment is not included in the summary statistics of Table 5).

The unsupervised pipeline begins with preprocessing (deterministic, except shuffling and random selection of development set), language model training with KenLM (Heafield, 2011) (deterministic), followed by phrase embedding training using phrase2vec (non-deterministic), and then embedding space mapping with VecMap (non-deterministic). Because performance on reruns of VecMap alone was stable while holding the rest of the system constant, we must conclude that the dramatic instability is caused by either a poor embedding initialization from phrase2vec/word2vec, or VecMap's inability to handle certain monolingual vector space configurations. Apparently, the initial formation of monolingual vector spaces dramatically affects VecMap's ability to converge to a good solution, which in turn results in highly variable downstream translation performance.

To observe the relationship between BLI accuracy and downstream BLEU score, we direct the reader to Figure 3, where BLI accuracy after the VecMap phase of experiments from Tables 3 and 4 are displayed in relation to the final BLEU score.

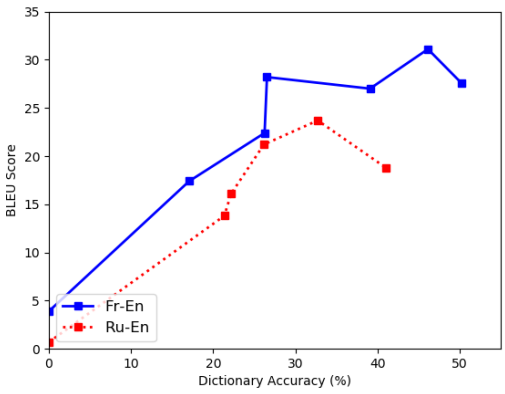


Figure 3: Relationship between bilingual lexicon induction accuracy after VecMap mapping, and final BLEU.

8 Discussion

Except in the Diff. Dom. condition, unsupervised MT performance for Fr-En is impressive and suggests that sentence alignment may not be required for successful MT under ideal conditions. Ru-En results are also impressive, but show that unsupervised MT still struggles when language pairs are dissimilar, especially when data amount is reduced.

The gap between Disjoint and Diff. Dom. conditions is perhaps the most striking result in our experiments. It suggests that one cannot naively collect monolingual corpora without considering their relative domain similarity; this may be a challenge in low-resource conditions, where there is less flexibility with data sources. Vulić et al. (2019) make a similar claim about unsupervised CLEs, stating "UNSUPERVISED methods are able to yield a good solution only when there is no domain mismatch and for the pair with two most similar languages (English-Spanish), again questioning their robustness and portability to truly low-resource and more challenging setups". Furthermore, the extremely poor results of Ne-En and Si-En reflect the reality of low-resource translation; the compound negative effects of language dissimilarity, domain mismatch between monolingual corpora, domain mismatch with the test set, and low amounts of low-quality data. It is the "worst of all worlds"—but reflects how current models might perform on the use cases for which they are needed. These challenges highlight the importance of evaluating unsupervised MT under varying realistic data conditions. Our evaluation is a step towards this goal, and identifies multiple areas for improvement.

A critical step in state-of-the-art unsupervised MT is methods for creating CLEs. Several authors have pointed out that "mapping" methods like VecMap assume that monolingual vector spaces are structurally similar, but that this "approximate isomorphism assumption" is increasingly tenuous as languages and domains diverge (Søgaard et al., 2018; Ormazabal et al., 2019; Glavaš et al., 2019; Vulić et al., 2019; Patra et al., 2019). Patra et al. (2019) find this for Fr-En and Ru-En specifically,

the languages examined in this work. Nakashole and Flauger (2018) argue that while linearity may hold within local "neighborhoods" of the vector space, the global mapping is non-linear. Søgaard et al. (2018) use their eigenvector similarity metric to show a strong correlation between vector space similarity and BLI performance. Analysis of the CLEs from our experiments demonstrate a relationship between BLI performance and downstream BLEU on the translation task. Coupled with our empirical evidence, the works cited in this section suggest that nonisometric vector spaces lead to poor quality translation.

Factors observed in our experiments that lead to lower quality translation can be attributable to a "weak isomorphism" between the monolingual vector spaces. Dissimilar languages means increasingly different distributional characteristics of words. Data from different domains naturally have different word frequencies and distributional characteristics, which become more pronounced as domains diverge. Because mapping methods rely on structural similarity of vector spaces, experiments using either UN or CC data alone had acceptable downstream performance, where as combining the datasets as source and target resulted in extremely poor translation. We observe the critical effect of word embedding initialization on BLI performance and downstream BLEU, suggesting that stochasticity during word embedding creation can cause resulting vector spaces to be more or less isomorphic. Finally, more data can give a more accurate distribution of words in comparison with the true distribution in the language, leading to a more realistic monolingual vector space. With less data, word embeddings are dependent on the smaller training sample, which may not match the test set or reflect true distributional properties of the language. Combining all of these negative factors likely leads to highly nonisomorphic monolingual embedding spaces, as demonstrated by the very poor Si-En and Ne-En results.

9 Conclusion & Future Work

Progress in unsupervised MT has been impressive, achieving performance near its supervised counterparts under some scenarios. That said, evaluating current approaches under broader settings and datasets reveals that unsupervised MT struggles in realistic low-resource scenarios. As stated by Lample et al. (2018b), "It's an open question whether there are more effective instantiations of these principles [underlying recent successes in fully unsupervised MT] or other principles altogether". In this work, we find that there is room for improvement to become robust to (1) dissimilar languages pairs, (2) dissimilar domains, (3) diverse datasets, and (4) the low-quality data of true low-resource languages—factors ubiquitous in low-resource language pairs for which unsupervised MT is intended. We find that (a) performance rapidly declines when source and target corpora are from different domains, and (b) stochasticity during word embedding training can dramatically affect downstream translation results. The latter is a yet unexplored research area. Future work should also evaluate pretraining methods in bilingual and multilingual training contexts.

Finally, we argue for extensive evaluation of unsupervised MT systems under varying data conditions to assess failure cases and encourage pursuit of promising paradigms. Doing so is a step towards solving the real-world problems of low-resource machine translation.

Acknowledgements

The authors would like to thank Mikel Artetxe for providing an implementation of his 2019 paper and thoughtful feedback, Matthew Francis-Landau, Cheng-I (Jeff) Lai, and Huda Khayrallah for support and advice. We also thank our anonymous reviewers and colleagues at Johns Hopkins University for helpful feedback.

This material is based upon work supported by the United States Air Force under Contract No. FA8750-19-C-0098. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Air Force and DARPA.

References

David Alvarez-Melis and Tommi Jaakkola. 2018. Gromov-Wasserstein alignment of word embedding spaces. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1881–1890, Brussels, Belgium. Association for Computational Linguistics.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018a. A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. In *Proceedings of the 56th Annual Meeting of*

the Association for Computational Linguistics (Volume 1: Long Papers), pages 789–798, Melbourne, Australia. Association for Computational Linguistics.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018b. Unsupervised statistical machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3632–3642, Brussels, Belgium. Association for Computational Linguistics.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2019. An effective approach to unsupervised machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 194–203, Florence, Italy. Association for Computational Linguistics.

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. 2018c. Unsupervised neural machine translation. In *Proceedings of the Sixth International Conference on Learning Representations*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems*, pages 7057–7067.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018. Word translation without parallel data. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Yerai Doval, Jose Camacho-Collados, Luis Espinosa-Anke, and Steven Schockaert. 2019. On the robustness of unsupervised and semi-supervised cross-lingual word embedding learning. *arXiv preprint arXiv:1908.07742*.

Pascale Fung. 1995. Compiling bilingual lexicon entries from a non-parallel English-Chinese corpus. In *Third Workshop on Very Large Corpora*.

Goran Glavaš, Robert Litschko, Sebastian Ruder, and Ivan Vulić. 2019. How to (properly) evaluate cross-lingual word embeddings: On strong baselines, comparative analyses, and some misconceptions. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 710–721, Florence, Italy. Association for Computational Linguistics.

Francisco Guzmán, Peng-Jen Chen, Myle Ott, Juan Pino, Guillaume Lample, Philipp Koehn, Vishrav Chaudhary, and Marc'Aurelio Ranzato. 2019. The FLORES evaluation datasets for low-resource machine translation: Nepali–English and Sinhala–English. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6098–6111, Hong Kong, China. Association for Computational Linguistics.

Aria Haghighi, Percy Liang, Taylor Berg-Kirkpatrick, and Dan Klein. 2008. Learning bilingual lexicons from monolingual corpora. In *Proceedings of ACL-08: HLT*, pages 771–779, Columbus, Ohio. Association for Computational Linguistics.

Mareike Hartmann, Yova Kementchedjhieva, and Anders Søgaard. 2019. Comparing unsupervised word translation methods step by step. In *Advances in Neural Information Processing Systems 32*, pages 6033–6043.

Kenneth Heafield. 2011. KenLM: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland. Association for Computational Linguistics.

Yedid Hoshen and Lior Wolf. 2018. Non-adversarial unsupervised word translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 469–478, Brussels, Belgium. Association for Computational Linguistics.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1700–1709, Seattle, Washington, USA. Association for Computational Linguistics.

Y. Kim, M. Graça, and H. Ney. 2020. When and why is unsupervised neural machine translation useless? *arXiv:2004.10581*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Philipp Koehn and Kevin Knight. 2000. Estimating word translation probabilities from unrelated mono-

lingual corpora using the em algorithm. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, page 711–715. AAAI Press.

Philipp Koehn and Kevin Knight. 2002. Learning a translation lexicon from monolingual corpora. In *Proceedings of the ACL-02 Workshop on Unsupervised Lexical Acquisition*, pages 9–16, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018a. Unsupervised machine translation using monolingual corpora only. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018b. Phrase-based & neural unsupervised machine translation. *CoRR*, abs/1804.07755.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *arXiv preprint arXiv:2001.08210*.

Tomas Mikolov, Quoc V Le, and Ilya Sutskever. 2013a. Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Ndapa Nakashole and Raphael Flauger. 2018. Characterizing departures from linearity in word translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 221–227, Melbourne, Australia. Association for Computational Linguistics.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 160–167, Sapporo, Japan. Association for Computational Linguistics.

Aitor Ormazabal, Mikel Artetxe, Gorka Labaka, Aitor Soroa, and Eneko Agirre. 2019. Analyzing the limitations of cross-lingual word embedding mappings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4990–4995, Florence, Italy. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Barun Patra, Joel Ruben Antony Moniz, Sarthak Garg, Matthew R. Gormley, and Graham Neubig. 2019. Bilingual lexicon induction with semi-supervision in non-isometric embedding spaces. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 184–193, Florence, Italy. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

Reinhard Rapp. 1995. Identifying word translations in non-parallel texts. In *Proceedings of the 33rd Annual Meeting on Association for Computational Linguistics*, ACL '95, page 320–322, USA. Association for Computational Linguistics.

Sujith Ravi and Kevin Knight. 2011. Deciphering foreign language. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 12–21, Portland, Oregon, USA. Association for Computational Linguistics.

Sebastian Ruder, Ivan Vulić, and Anders Søgaard. 2019. A survey of cross-lingual word embedding models. *Journal of Artificial Intelligence Research*, 65:569–631.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Anders Søgaard, Sebastian Ruder, and Ivan Vulić. 2018. On the limitations of unsupervised bilingual dictionary induction. In *Proceedings of the 56th Annual Meeting of the Association for Computational*

Linguistics (Volume 1: Long Papers), pages 778–788, Melbourne, Australia. Association for Computational Linguistics.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. Mass: Masked sequence to sequence pre-training for language generation. In *ICML*.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2214–2218, Istanbul, Turkey. European Language Resources Association (ELRA).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Ivan Vulić, Goran Glavaš, Roi Reichart, and Anna Korhonen. 2019. Do we really need fully unsupervised cross-lingual embeddings? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4407–4418, Hong Kong, China. Association for Computational Linguistics.

Krzysztof Wołk and Krzysztof Marasek. 2014. Building subject-aligned comparable corpora and mining it for truly parallel sentence pairs. *Procedia Technology*, 18:126–132.

Zhen Yang, Wei Chen, Feng Wang, and Bo Xu. 2018. Unsupervised neural machine translation with weight sharing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 46–55, Melbourne, Australia. Association for Computational Linguistics.

Michał Ziemski, Marcin Junczys-Dowmunt, and Bruno Pouliquen. 2016. The united nations parallel corpus v1.0. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3530–3534, Portorož, Slovenia. European Language Resources Association (ELRA).

Language Models not just for Pre-training: Fast Online Neural Noisy Channel Modeling

Shruti Bhosale[△] **Kyra Yee**[†▽] **Sergey Edunov**[△] **Michael Auli**[△]
[△] Facebook AI Research, Menlo Park, CA, USA
[▽] Twitter Cortex, San Francisco, CA, USA

Abstract

Pre-training models on vast quantities of unlabeled data has emerged as an effective approach to improving accuracy on many NLP tasks. On the other hand, traditional machine translation has a long history of leveraging unlabeled data through noisy channel modeling. The same idea has recently been shown to achieve strong improvements for neural machine translation. Unfortunately, naïve noisy channel modeling with modern sequence to sequence models is up to an order of magnitude slower than alternatives. We address this issue by introducing efficient approximations to make inference with the noisy channel approach as fast as strong ensembles while increasing accuracy. We also show that the noisy channel approach can outperform strong pre-training results by achieving a new state of the art on WMT Romanian-English translation.

1 Introduction

Unlabeled data has been leveraged in many ways in natural language processing including back-translation (Bojar and Tamchyna, 2011; Sennrich et al., 2016b; Edunov et al., 2018), self-training (He et al., 2020), or language model pre-training which led to improvements in many natural language tasks (Devlin et al., 2019). While pre-training has achieved impressive results on tasks where labeled data is limited, improvements in settings with abundant labeled data are modest (Raffel et al., 2020) with controlled studies showing a clear trend of diminishing returns as the amount of training data increases (Edunov et al., 2019).

In this paper, we focus on noisy channel modeling for text generation tasks, a classical technique from the statistical machine translation literature which had been the workhorse of text generation tasks for decades before the arrival of neural sequence to sequence models (Brown et al., 1993; Koehn et al., 2003). Unlike pre-training approaches, this approach is very effective irrespective of the amount of labeled data: since a recent revival (Yu et al., 2017; Yee et al., 2019), it has been an important part in the winning entries of several high resource language pairs at WMT 2019 (Ng et al., 2019), improving over strong ensembles that used 500M back-translated sentences. At the low resource WAT 2019 machine translation competition, noisy channel modeling was also a key factor for the winning entry (Chen et al., 2019).

Noisy channel modeling turns text generation on the head: instead of modeling an output sequence given an input, Bayes' rule is applied to model the input given the output, via a backward sequence to sequence model which is combined with the prior probability of the output, typically a language model. This enables the effective use of strong language models trained on large amounts of unlabeled data. The role of the backward model, or the channel model, is to validate outputs preferred by the language model with respect to the input.

A straightforward way to use language models is to pair them with standard sequence to sequence models (Gülçehre et al., 2015; Stahlberg et al., 2018). However, this does not address *explaining away effects* under which modern neural sequence models still suffer (Klein and Manning, 2001; Li et al., 2019). As a consequence, models are susceptible to producing fluent outputs that are unrelated to the input (Li et al., 2019). The noisy channel approach explicitly addresses this via the channel model.

However, a major obstacle to efficient noisy channel modeling is that generating outputs is much slower than decoding from a standard sequence to sequence model. We address this issue by introducing several simple yet highly ef-

† Work done while at Facebook during a Facebook AI Residency.

Proceedings of the 5th Conference on Machine Translation (WMT), pages 584–593
Online, November 19–20, 2020.

fective approximations which increase the speed of noisy channel modeling by an order of magnitude to make it practical. This includes smaller channel models as well as scoring only a subset of the channel model vocabulary. Experiments on WMT English-Romanian translation show that noisy channel modeling can outperform recent pre-training results. Moreover, we show that noisy channel modeling benefits much more from larger beam sizes than strong pre-training methods.

2 The Noisy Channel Approach

We assume a sequence to sequence task that takes the input x to predict the output y. A standard sequence to sequence model directly estimates the probability $p(y|x)$, referred to as a *direct model*. On the other hand, the noisy channel approach applies Bayes' rule to model $p(y|x) = p(x|y)p(y)/p(x)$ where $p(x|y)$ predicts the source x given the target y and is referred to as the *channel model*, $p(y)$ is a language model over the target y, and $p(x)$ is generally not modeled since it is constant for all y.

Yee et al. (2019) use Transformer models to parameterize the direct model, the channel model and the language model. Similar to Yu et al. (2017), they use the following linear combination of the channel model, the language model as well as the direct model for decoding:

$$\frac{1}{t}\log p(y|x) + \frac{\lambda_1}{s}\log p(x|y) + \frac{\lambda_2}{s}\log p(y) \quad (1)$$

where t is the length of the output prefix y, s is the length of the input sequence, and λ_1, λ_2 are hyperparameters.

Exact noisy channel model scoring with neural networks during decoding is prohibitively expensive since it requires a separate forward computation with the channel model for every token in the target vocabulary. To side step this issue, Yu et al. (2017) propose the following approximations to beam search with beam width k_1: determine the k_2 highest scoring extensions of each beam according to the direct model, then score the resulting $k_1 \times k_2$ partial candidates by the direct model, the channel model and the language model using the linear combination in Equation 1. Finally, this set is pruned to beam size k_1.

Despite this approximation, noisy channel decoding is still significantly slower than decoding with the direct model alone as shown in Figure 1.

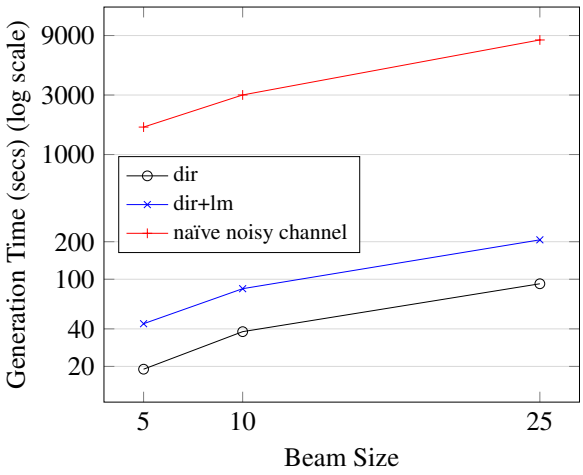


Figure 1: Speed of decoding with a direct model (`dir`), direct model with language model (`dir + lm`) and a naïve noisy channel approach without fast approximations or optimizations. The latter is very slow compared to the direct model. Results are based on generation with the fastest batch size for each setting with beam 5 on newstest2016 De-En (cf. §4.1).

The reason for this is that the channel model repeatedly scores the entire input sequence at each time-step and this is done k_2 times for each beam. Specifically, both the direct model and the language model compute $k_1 \times V$ scores at each time-step in order to make a decoding decision for each target token during beam search, where V denotes the vocabulary size which we assume to be similar between the input and output. In contrast, the channel model computes $k_1 \times k_2 \times S \times V$ scores for each target token, where S is the maximum source sequence length. This adds substantial compute and memory overhead, to the extent that the batch size at decoding often needs to be substantially reduced. This leads to slower inference on GPUs since less computation can be parallelized.

3 Fast Noisy Channel Modeling

Naïve online noisy channel modeling is significantly slower than standard direct models. In this section, we present approximations to make noisy channel modeling substantially faster.

3.1 Reducing Channel Model Size

Prior work on neural noisy channel used channel models which were of the same size as the direct model (Yu et al., 2017; Yee et al., 2019). The most recent work uses standard Transformer models (Yee et al., 2019; Ng et al., 2019; Yu et al., 2020). In this study, we hypothesize that the primary role

of the channel model is to avoid *explaining away effects* by the language model. This primarily entails assigning low scores to unrelated outputs, which may not require a very powerful model. In this case, we may be able to substantially decrease the size of the channel model at only a small loss in accuracy.

Recent work demonstrates that direct models with shallow decoders can give comparable accuracy, while being faster at inference time, compared to models with deep decoders (Wu et al., 2019; Elbayad et al., 2020; Kasai et al., 2020; Fan et al., 2020). This is particularly attractive for direct models for which the decoder network accounts for most of the wall time during inference but the dynamics for channel models are different: the channel model repeatedly scores the entire input sequence given progressively larger target prefixes. Unlike for direct models, there is no straightforward way to reuse the encoder output between time-steps, and we opt to recompute the entire encoder and decoder of the channel model at every target time-step. Since the input sequence is given, channel model computation can be batched over all tokens in the target prefix and the input sequence. This implies that we are free to adjust both the encoder and decoder depth.

We pursue two strategies to reduce model size: first, we progressively reduce the model dimension of the `base` Transformer architecture, by first halving the model dimension from 512 to 256, as well as the feed forward dimension from 2048 to 1024 for the `half` model. The smallest configuration uses a model dimension of just 32 and a feed forward dimension of 128 (denoted as `16th` model). Second, we consider models with only a single encoder block and a single decoder block. These models have a postfix `_1_1`, e.g., `16th_1_1`. Table 1 shows the various model sizes as well as accuracy on the development set, newstest2016.

	Parameters (M)	BLEU
big	282.7	38.0
big_1_1	93.8	34.1
base	72.1	36.7
base_1_1	23.6	31.2
half	25.1	33.6
half_1_1	15.8	27.4
quarter	9.8	28.4
quarter_1_1	7.5	22.0
16th	2.8	15.9
16th_1_1	2.7	10.0

Table 1: Smaller channel models in terms of number of total parameters as well as BLEU (avg. over 3 seeds) on the development set. All models have six blocks each in the encoder and the decoder, except for models ending in "_1_1" which have only a single block in the encoder and the decoder.

3.2 Reducing the Output Vocabulary

During online noisy channel decoding, we need to allocate memory for a large number of channel model output probabilites ($k_1 \times k_2 \times S \times V$, as explained in § 2). This substantially reduces the maximum possible batch size in order to prevent running out of memory while decoding on GPUs. A small batch size prevents the full utilization of parallel computation on GPUs, particularly, when the channel model is relatively small: some of our channel models have an embedding dimension of just 32.

To address this issue, we make use of the fact that we know exactly which input tokens need to be scored (since the input sequence is given) instead of computing probabilities for the entire vocabulary. This is similar to vocabulary reduction techniques used for early neural sequence to sequence models, and it is particularly convenient since we know exactly which tokens are in the input sequence (Mi et al., 2016; L'Hostis et al., 2016).

Similar to prior work on vocabulary reduction, we found it useful to not just score the input words but also a subset of the most frequent words in the vocabulary. Specifically, for each batch, we enumerate all input word types, add the 500 most frequent types and then compute output probabilities for this subset with the channel model. The number of output probabilities calculated is typically at least one order of magnitude smaller than the full vocabulary, as shown in § 5.4.1.

This approach substantially reduces the memory footprint of small channel models and enables the use of much larger batch sizes which leads to faster inference as we will see in § 5.

3.3 Reducing the Number of Candidates

We also study the effect of reducing the number of next token candidates k_2 scored for each beam at each step of beam search. This reduces the computation as well as memory overhead of channel model scoring.

4 Experimental Setup

4.1 Datasets

We consider two datasets for our experiments: For German-English (De-En), we train on WMT'19 training data. Following (Ng et al., 2019), we apply language identification filtering (Lui and Baldwin, 2012) and remove sentences longer than 250 tokens as well as sentence pairs with a source/target length ratio exceeding 1.5. This results in 26.8M sentence pairs. We validate on newstest2016 and test on newstest2014, newstest2015, newstest2017, and newstest2018. For all models, the source vocabulary is a 24K byte pair encoding (BPE; Sennrich et al., 2016) learned on the source portion of the bitext. For the target side, we use the vocabulary of the language model (§4.2) so that both models score the exact same units during beam search.

For Romanian-English (Ro-En), we train on WMT'16 training data, comprising 612K sentence pairs, validate on newsdev2016 and test on newstest2016. We learn a joint BPE vocabulary of 18K types on the bitext training data which is used for both the source and target. Different to German-English, we learn a joint BPE vocabulary to enable sharing the source and target embeddings which we found to perform better for Romanian-English in early experiments.

4.2 Language Models

For German-English, we train a sentence-level English Transformer language model with 16 layers and Transformer-Big architecture (Vaswani et al., 2017; Radford et al., 2018). The model is trained on de-duplicated English Newscrawl data from 2007-2018 comprising 186 million sentences or 4.5B words after normalization and tokenization. We use a BPE vocabulary of 24K types learned on this data. For Romanian-English translation, we train a similar English Transformer language model that uses the joint BPE vocabulary learned on the Romanian-English bitext. The latter enables the LM to score the exact same units as the sequence to sequence model during beam search.

We train a sentence-level Romanian Transformer language model with 16 layers and Transformer-Big architecture. The model is trained on de-duplicated Romanian CommonCrawl data consisting of 623M sentences or 21.7B words after normalization and tokenization (Conneau et al., 2019; Wenzek et al., 2020).

The German-English bitext training data as well as the language model training data are preprocessed with the Moses tokenizer (Koehn et al., 2007). We normalize punctuation and remove non-printing characters. Romanian-English data is preprocessed following Sennrich et al. (2016a) by applying Moses tokenization and special normalization for Romanian text.[1]

4.3 Translation Models

For De-En, we use the Transformer-Big architecture for the direct model. We do not share encoder and decoder embeddings since the source and target vocabularies are different. For channel models, operating from English to German, we consider different variants (§3.1, Table 1) to better understand the speed-accuracy trade-off of decreasing the capacity of channel models.

For Ro-En and En-Ro with bitext only, the direct and channel models use a Transformer-Base architecture. For Ro-En with backtranslation, the direct and channel models use a Transformer-Big architecture. We share the encoder and decoder embeddings since the source and target vocabularies are the same and because this improved accuracy.

4.4 Online Noisy Channel Decoding Setup

In order to set weights for the linear combination of model scores (Equation 1), we randomly sample a set of hyperparameters and evaluate each configuration on the development set (Yee et al., 2019; Ng et al., 2019). Hyperparameters are sampled within the interval $[0, 2]$, For direct models (`dir`), we sample ten random weights for the length penalty. For direct models combined with language models (`dir + lm`), we evaluate 100 randomly sampled configurations for the length penalty and the language model weight (λ_2). For direct models combined with language models and channel models (`dir + lm + ch`), we evaluate 1000 configurations of the length penalty, the language model weight (λ_2) and the channel model weight (λ_1). We use 16-bit floating point precision (Ott et al., 2018, 2019) for decoding with the online noisy channel setup.

Accuracy is measured via sacreBLEU (Post, 2018) for WMT German-English. We report the average BLEU of the newstest2014-2015 and newstest2017-2018 test sets, averaged over 3 random seeds for model weight initialization. Speed

[1] https://github.com/rsennrich/wmt16-scripts/tree/master/preprocess

	Total Params (M)	BLEU	Time (s)
Ensembles			
dir	283	38.8	20
2 dir	565	39.3	40
3 dir	848	39.5	59
Ensembles + LMs			
dir + lm	539	39.7	44
2 dir + lm	822	40.2	65
3 dir + lm	1104	40.3	84
Noisy Channel Modeling (Yee et al., 2019)			
dir + lm + big	822	40.5	550
Fast Noisy Channel Modeling (This work)			
dir + lm + 16th_1_1	542	40.2	56
dir + lm + base_1_1	574	40.5	92
2 dir + lm + 16th_1_1	824	40.5	76
2 dir + lm + base_1_1	857	40.8	111
3 dir + lm + 16th_1_1	1107	40.6	93
3 dir + lm + base_1_1	1140	41.0	131

Table 2: Fast noisy channel modeling is more accurate than ensembles at comparable speed and the two methods are additive. All results use beam size 5, batch sizes for each configuration are optimized and BLEU is averaged over news2014, news2015, news2017 and news2018 of WMT German to English.

	Channel Model Params (M)	BLEU	Time (s)
dir + lm + big	283	40.3	472
dir + lm + base	72	40.4	202
dir + lm + half	25	40.5	132
dir + lm + quarter	10	40.4	102
dir + lm + 8th	6	40.3	89
dir + lm + 16th	3	40.2	70
dir + lm + big_1_1	94	40.5	160
dir + lm + base_1_1	24	40.5	92
dir + lm + half_1_1	16	40.4	72
dir + lm + quarter_1_1	8	40.2	63
dir + lm + 8th_1_1	5	40.3	60
dir + lm + 16th_1_1	3	40.2	56

Table 3: Smaller channel models perform similarly for the standard beam size of 5. We exploit this fact to speed up noisy channel decoding.

is measured by the generation time (averaged over 3 trials) in seconds on the German-English newstest2016 test set on a 32GB Volta V100 GPU using 16-bit floating point precision (Ott et al., 2018, 2019). Unless otherwise specified, the beam size is 5, and the number of candidates for noisy channel model scoring per beam is $k_2 = 10$, unless otherwise specified. Generation times are based on a tuned batch size for each model configuration. We select the batch size within $(1, 10, 25, 50, 75, 100, 125, 150, 200, 300)$ that fits in memory and results in the fastest generation time.

5 Results

5.1 Fast Noisy Channel Modeling

In the first experiment, we evaluate the speed and accuracy of fast noisy channel decoding (§ 3) and compare to the naïve version without approximations (Yee et al., 2019). As additional baselines, we consider a single direct model (`dir`), ensembling two direct models (`2 dir`) and three direct models (`3 dir`), as well as adding a language model to each (`lm`). As channel models, we consider a `big` Transformer, a `base` Transformer, as well as a variant with model dimension of only 32 which is 1/16th of the model dimension of a `base` Transformer with a single layer in the encoder and decoder each (`16th_1_1`), totaling just 2.7M parameters. For fast noisy channel decoding, we reduce the channel model output vocabulary (§3.2) and set $k_2 = 3$; we ablate these choices in § 5.4.

Table 2 shows that the approximations we introduce to make noisy channel decoding fast also achieve similar accuracy (40.5 BLEU) to the much slower noisy channel approach of (Yee et al., 2019), while being about six times faster at inference time.

Table 2 also shows that `dir + lm + 16th_1_1` is 0.7 BLEU score better than `3 dir` at a similar decoding speed. Thus, using a small channel model and a language model with online noisy channel decoding is a better strategy than ensembling 3 direct models. Noisy channel decoding is also complementary to ensembling direct models: `3 dir + lm + base_1_1` improves by 0.7 BLEU compared to `3 dir + lm`.

Table 3 compares fast noisy channel decoding with different channel model sizes. Generally, smaller channel models are only slightly less accurate than larger models while being significantly faster than their larger counterparts. For example,

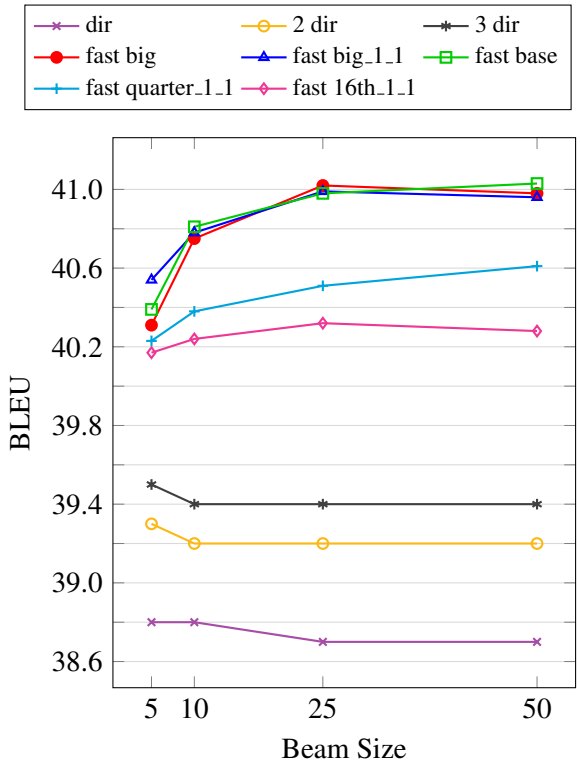


Figure 2: BLEU of fast online noisy channel decoding with different channel models when beam size is increased (compared to ensemble baselines). BLEU is averaged over news2014, news2015, news2017 and news2018 of WMT De-En.

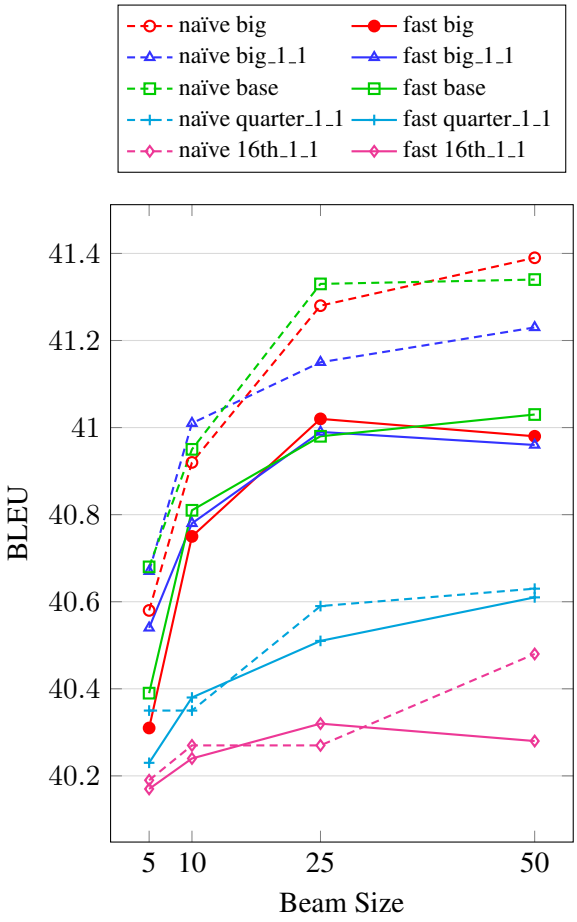


Figure 3: BLEU of fast and naïve online noisy channel decoding with different channel models sizes when beam size is increased. BLEU is averaged over news2014, news2015, news2017 and news2018 of WMT De-En.

`16th_1_1` is over eight times faster than `big` and achieves nearly the same accuracy.

This observation is in line with the hypothesis that the primary role of the channel model is to tie back the language model generations to the input. We exploit the fact that small channel models work well to make noisy channel decoding very fast.

5.2 Noisy Channel Decoding with Larger Beam Sizes

So far we used a standard beam size of five to enable fast decoding. However, previous work found that noisy channel modeling benefits more from larger beam sizes than other methods (Yee et al., 2019). Next, we evaluate whether our efficiency improvements still enable good performance with larger beam sizes.

Figure 2 shows that for beam size 5, most channel models perform comparably. Larger models are slightly better but overall they are in a similar ball park. As the beam size increases, larger channel models do achieve better accuracy. However, there is no difference between a single layer big model (`big_1_1`) and a six layer version (`big`). As observed in previous work (Yee et al., 2019), the direct model and the direct ensembles (`dir`, `2 dir`, `3 dir`) do not benefit from larger beam sizes.

Next, we compare fast noisy channel decoding and naïve noisy channel decoding at larger beam sizes. As shown in Figure 4, the naïve approach is much slower. Fast approximations to noisy channel decoding scale much better in terms of speed as the beam size increases. Figure 3 compares the accuracy of fast noisy channel decoding at larger beam sizes with that of naïve noisy channel decoding. Using the `big` and `big_1_1` channel models gives the best performance across all beam sizes for naïve noisy channel decoding. With fast noisy channel decoding, we see an average drop of 0.3 BLEU and 0.2 BLEU for `big` and `big_1_1` respectively. On the other hand, for smaller channel models, the difference between naïve and fast noisy

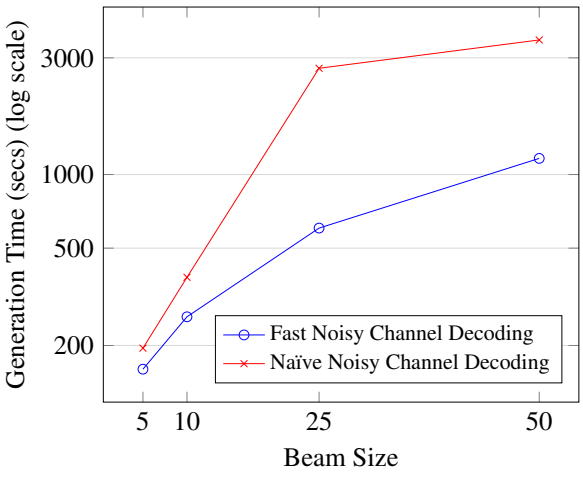


Figure 4: With larger beam sizes, the speed of fast approximations for noisy channel decoding scales much better than that of naïve noisy channel decoding. Results are based on generation using the `big_1_1` channel model with the fastest batch size for each setting with beam 5 on newstest2016 De-En.

channel decoding is generally smaller.

5.3 Results on WMT Romanian-English

Next, we evaluate noisy channel modeling on WMT Romanian-English translation (Ro-En and En-Ro) which is a low resource setup compared to WMT German-English. We also compare to a recently introduced pre-training approach, mBART. The mBART model is pre-trained to denoise input sentences in multiple languages, followed by fine-tuning on the bitext (Liu et al., 2020). Following their setup for En-Ro evaluation, we apply Moses tokenization and normalize diacritics for Romanian (Sennrich et al., 2016a), and use tokenized BLEU. For Ro-En, we use SacreBLEU (Post, 2018).

Table 4 shows that noisy channel decoding with a wide beam can outperform multilingual pre-training (mBART) across the board. Large beams are not helpful for generation with mBART. Compared to the direct model, noisy channel decoding improves by 2.7/3.1 BLEU on En-Ro and Ro-En respectively, and increasing the beam size gives gains of 4.5/4.5 BLEU.

We also study the performance of noisy channel decoding on Romanian-English with back-translated data generated using unrestricted sampling (Edunov et al., 2018).[2] As compared to mBART02 (Liu et al., 2020), the previous state-of-the-art result on Romanian-English with back-translation, we achieve a 0.5 BLEU improvement. We use a similar number of total model parameters, but much less monolingual English data. Our English language model is trained on 4.5B tokens, while mBART02 uses 66B tokens of English and Romanian monolingual data.

Finally, Table 5 shows that fast approximations and smaller channel models achieve similar performance but much higher speed compared to naïve noisy channel decoding on WMT Romanian-English with back-translation. Fast noisy channel decoding with `base_1_1` achieves comparable accuracy as mBART02 at slightly faster generation time with beam size 5.

5.4 Ablations

In this section we focus on some of the design choices we made to speed up noisy channel decoding. We measure the impact on speed and accuracy when reducing the output vocabulary size of the channel model, and reducing the number of beam candidates scored by the channel model.

5.4.1 Reducing the Output Vocabulary

In the next experiment, we compare the speed of using the full output vocabulary for the channel model to a reduced version. Specifically, we reduce the vocabulary by selecting all source tokens in the batch as well as the most frequent 500 tokens in the training data (see § 3.2). We tune each setup by selecting the fastest batch size based on a sweep over different batch sizes $(1, 10, 25, 50, 75, 100, 125, 150, 200, 300)$.

Table 6 shows that generating channel model scores for a small subset of the source vocabulary results in a small accuracy of up to 0.3 BLEU, but often less, while substantially increasing speed by 40-65% for single layer channel models and by 20-55% for other channel models. `base_1_1` with a small vocabulary is nearly ten times faster than the approach proposed in Yee et al. (2019) (channel model size `big`), with a slight decline in accuracy.

The average vocabulary size used for scoring the channel model is around 1050, as compared to full source vocabulary size of 28,048. This leads to a large reduction in memory consumption and enables fitting larger batches into memory.

[2]The monolingual English data used for backtranslation comes from `http://data.statmt.org/rsennrich/wmt16_backtranslations/` (Sennrich et al., 2016c).

	mono tokens Ro-En (B)	mono tokens En-Ro (B)	En-Ro	Ro-En	Ro-En +BT
mBART02	66	66	38.5	38.5	39.9
mBART02 (beam=50)	66	66	-	-	39.9
dir	-	-	34.6	34.6	38.4
dir + lm	4.5	22	35.4	35.9	38.7
dir + lm + big	4.5	22	37.3	37.7	39.6
dir + lm + big (beam=50)	4.5	22	**39.1**	**39.1**	**40.4**

Table 4: BLEU of noisy channel decoding on the Romanian-English newstest2016 test set with bitext-only as well as with backtranslation (BT) compared to mBART (Liu et al., 2020). We also show the total amount of monolingual data used by each method in billions of tokens.

	BLEU	Time (s)
mBART02	39.9	93
mBART02 (beam=50)	39.9	754
dir	38.4	19
Noisy Channel Modeling (Yee et al., 2019)		
dir + lm + big	39.6	1178
dir + lm + big (beam=50)	40.4	12554
Fast Noisy Channel Modeling		
dir + lm + base_1_1	39.8	82
dir + lm + base_1_1 (beam=50)	40.3	631

Table 5: Speed and accuracy on Romanian-English (Ro-En) with backtranslation. Fast noisy channel decoding using `base_1_1` achieves similar accuracy to mBART02 while being faster (beam=5). BLEU is measured on newstest2016 and generation time is measured on newsdev2016.

dir+ch+lm (beam=5)	Full Source Vocab		Small Source Vocab	
	BLEU	Time (s)	BLEU	Time (s)
big	40.6	1656	40.3	1355
base	40.7	854	40.4	516
half	40.6	450	40.5	299
quarter	40.5	359	40.4	212
8th	40.3	324	40.3	178
16th	40.1	264	40.2	118
big_1_1	40.7	543	40.5	339
base_1_1	40.5	336	40.5	169
half_1_1	40.3	264	40.4	117
quarter_1_1	40.4	238	40.2	95
8th_1_1	40.1	223	40.3	87
16th_1_1	40.2	209	40.2	74

Table 6: Comparison of accuracy (BLEU) and speed of online noisy channel decoding with and without the small output vocabulary approximation for different channel model sizes. Note we use $k_2 = 10$ for this ablation. BLEU is averaged over news2014, news2015, news2017 and news2018 of WMT De-En and generation time is on news2016.

5.4.2 Reducing the Number of Candidates

For each beam in each step of beam search, we need to make a choice about how many candidates k_2 we re-score with noisy channel modeling. Yee et al. (2019) re-scored $k_2 = 10$ candidates for each beam at each step. We sweep over different values of k_2 to understand the speed-accuracy trade-off associated with the choice of k_2. Table 7 shows that smaller values for k_2 are as accurate and much faster for beam size 5.

6 Conclusion

We introduced a number of approximations which greatly speed up noisy channel modeling for neural sequence to sequence models. This includes using channel models which are a fraction of the size of commonly used sequence to sequence models, pruning most of the channel model output vocabulary, and reducing the number of beam candidates scored by the channel model.

Our approximations are simple, yet, highly effective and enable comparable inference speed to ensembles of direct models while delivering higher accuracy. Our experiments show that noisy channel modeling can outperform pre-training approaches by being able to better exploit wider beams. Moreover, this is achieved while using a smaller amount of monolingual data.

k_2	BLEU	Time (s)
2	40.4	76
3	40.5	88
5	40.4	124
10	40.5	168

Table 7: Smaller number of rescoring candidates k_2 per beam are as accurate and much faster than larger values of k_2 for fast noisy channel decoding using `base_1_1` with beam 5. BLEU is averaged over news2014, news2015, news2017 and news2018 of WMT De-En and generation time is on news2016.

References

Ondrej Bojar and Ales Tamchyna. 2011. Improving translation model by monolingual data. In *Proc. of WMT*.

Peter F Brown, Vincent J Della Pietra, Stephen A Della Pietra, and Robert L Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational linguistics*.

Peng-Jen Chen, Jiajun Shen, Matt Le, Vishrav Chaudhary, Ahmed El-Kishky, Guillaume Wenzek, Myle Ott, and Marc'Aurelio Ranzato. 2019. Facebook ai's wat19 myanmar-english translation task submission. *WAT 2019*, page 112.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proc. of NAACL*.

Sergey Edunov, Alexei Baevski, and Michael Auli. 2019. Pre-trained language model representations for language generation. In *Proc. of NAACL*.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Proc. of EMNLP*.

Maha Elbayad, Jiatao Gu, Edouard Grave, and Michael Auli. 2020. Depth-adaptive transformer. In *Proc. of ICLR*.

Angela Fan, Edouard Grave, and Armand Joulin. 2020. Reducing transformer depth on demand with structured dropout. In *Proc. of ICLR*.

Çaglar Gülçehre, Orhan Firat, Kelvin Xu, Kyunghyun Cho, Loïc Barrault, Huei-Chi Lin, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2015. On using monolingual corpora in neural machine translation. *arXiv*, abs/1503.03535.

Junxian He, Jiatao Gu, Jiajun Shen, and Marc'Aurelio Ranzato. 2020. Revisiting self-training for neural sequence generation. In *Proc. of ICLR*.

Jungo Kasai, Nikolaos Pappas, Hao Peng, James Cross, and Noah A. Smith. 2020. Deep encoder, shallow decoder: Reevaluating the speed-quality tradeoff in machine translation. *arXiv*, abs/2006.10369.

Dan Klein and Christopher Manning. 2001. Conditional structure versus conditional estimation in nlp. In *Proc. of EMNLP*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*, pages 177–180. Association for Computational Linguistics.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proc. of NAACL*.

Gurvan L'Hostis, David Grangier, and Michael Auli. 2016. Vocabulary Selection Strategies for Neural Machine Translation. *arXiv*, abs/1610.00072.

Margaret Li, Stephen Roller, Ilia Kulikov, Sean Welleck, Y-Lan Boureau, Kyunghyun Cho, and Jason Weston. 2019. Don't say that! making inconsistent dialogue unlikely with unlikelihood training. *arXiv*.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *arXiv preprint arXiv:2001.08210*.

Marco Lui and Timothy Baldwin. 2012. langid. py: An off-the-shelf language identification tool. In *Proceedings of the ACL 2012 system demonstrations*, pages 25–30.

Haitao Mi, Zhiguo Wang, and Abe Ittycheriah. 2016. Vocabulary manipulation for neural machine translation. In *Proc. of ACL*.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook fair's wmt19 news translation task submission. In *Proc. of WMT*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proc. of NAACL System Demonstrations*.

Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. 2018. Scaling neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 1–9.

Matt Post. 2018. A call for clarity in reporting bleu scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Edinburgh neural machine translation systems for wmt 16. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 371–376.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Improving neural machine translation models with monolingual data. In *Proc. of ACL*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016c. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016d. Neural machine translation of rare words with subword units. In *Proc. of ACL*.

Felix Stahlberg, James Cross, and Veselin Stoyanov. 2018. Simple fusion: Return of the language model. In *Proc. of WMT*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Édouard Grave. 2020. Ccnet: Extracting high quality monolingual datasets from web crawl data. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 4003–4012.

Felix Wu, Angela Fan, Alexei Baevski, Yann N. Dauphin, and Michael Auli. 2019. Pay less attention with lightweight and dynamic convolutions. In *Proc. of ICLR*.

Kyra Yee, Yann N Dauphin, and Michael Auli. 2019. Simple and effective noisy channel modeling for neural machine translation. In *Proc. of NAACL*.

Lei Yu, Phil Blunsom, Chris Dyer, Edward Grefenstette, and Tomás Kociský. 2017. The neural noisy channel. In *Proc. of ICLR*.

Lei Yu, Laurent Sartran, Wojciech Stokowiec, Wang Ling, Lingpeng Kong, Phil Blunsom, and Chris Dyer. 2020. Better document-level machine translation with bayes' rule. *TACL*.

Towards Multimodal Simultaneous Neural Machine Translation

Aizhan Imankulova* **Masahiro Kaneko*** **Tosho Hirasawa*** **Mamoru Komachi**
Tokyo Metropolitan University
6-6 Asahigaoka, Hino, Tokyo 191-0065, Japan
{imankulova-aizhan, kaneko-masahiro, hirasawa-tosho}@ed.tmu.ac.jp
komachi@tmu.ac.jp

Abstract

Simultaneous translation involves translating a sentence before the speaker's utterance is completed in order to realize real-time understanding in multiple languages. This task is significantly more challenging than the general full sentence translation because of the shortage of input information during decoding. To alleviate this shortage, we propose multimodal simultaneous neural machine translation (MSNMT), which leverages visual information as an additional modality. Our experiments with the Multi30k dataset showed that MSNMT significantly outperforms its text-only counterpart in more timely translation situations with low latency. Furthermore, we verified the importance of visual information during decoding by performing an adversarial evaluation of MSNMT, where we studied how models behaved with incongruent input modality and analyzed the effect of different word order between source and target languages.

1 Introduction

Simultaneous translation is a natural language processing (NLP) task in which translation begins before receiving the whole source sentence. It is widely used in international summits and conferences where real-time comprehension is one of the essential aspects. Simultaneous translation is already a difficult task for human interpreters because the message must be understood and translated while the input sentence is still incomplete, especially for language pairs with different word orders (e.g. SVO-SOV) (Seeber, 2015). Consequently, simultaneous translation is more challenging for machines. Previous works attempt to solve this task by predicting the sentence-final verb (Grissom II et al., 2014), or predicting unseen syntactic constituents (Oda et al., 2015). Given the difficulty of predicting future inputs based on existing limited inputs, Ma et al. (2019) proposed a simple simultaneous neural machine translation (SNMT) approach `wait-k` which generates the target sentence concurrently with the source sentence, but always `k` tokens behind, satisfying low latency requirements.

However, previous approaches solve the given task by solely using the text modality, which may be insufficient to produce a reliable translation. Simultaneous interpreters often consider various additional information sources such as visual clues or acoustic data while translating (Seeber, 2015). Therefore, we hypothesize that using supplementary information, such as visual clues, can also be beneficial for simultaneous machine translation.

To this end, we propose Multimodal Simultaneous Neural Machine Translation (**MSNMT**) that supplements the incomplete textual modality with visual information, in the form of an image. It will predict still missing information to improve translation quality during the decoding process. Our approach can be applied in various situations where visual information is related to the content of speech such as presentations with slides (e.g. TED Talks[1]) and news video broadcasts[2]. Our experiments show that the proposed MSNMT method achieves higher translation accuracy than the SNMT model that does not use images by leveraging image information. To the best of our knowledge, we are the first to propose the incorporation of visual information to solve the problem of incomplete text information in SNMT.

The main contributions of our research are as follows. We propose to combine multimodal and simultaneous NMT, therefore, discovering cases where such multimodal signals are beneficial for

*These authors contributed equally to this paper

[1]https://interactio.io/

[2]https://www.a.nhk-g.co.jp/bilingual-english/broadcast/nhk/index.html

Proceedings of the 5th Conference on Machine Translation (WMT), pages 594–603
Online, November 19–20, 2020.

the end-task. Our MSNMT approach brings significant improvement in simultaneous translation quality by enriching incomplete text input information using visual clues. As a result of a thorough analysis, we conclude that the proposed method is able to predict tokens that have not appeared yet for source-target language pairs with different word order (e.g. English→Japanese). By providing an adversarial evaluation, we showed that the models indeed utilize visual information.

2 Related Work

For simultaneous translation, it is crucial to predict the words that have not appeared yet. For example, it is important to distinguish nouns in SVO-SOV translation and verbs in SOV-SVO translation (Ma et al., 2019). SNMT can be realized with two types of policy: fixed and adaptive policies (Zheng et al., 2019b). Adaptive policy decides whether to wait for another source word or emit a target word in one model. Previous models with adaptive policies include explicit prediction of the sentence-final verb (Grissom II et al., 2014; Matsubara et al., 2000) and unseen syntactic constituents (Oda et al., 2015). Most dynamic models with adaptive policies (Gu et al., 2017; Dalvi et al., 2018; Arivazhagan et al., 2019; Zheng et al., 2019a,c, 2020) have the advantage of exploiting input text information as effectively as possible due to the lack of such information in the first place. Meanwhile, Ma et al. (2019) proposed a simple `wait-k` method with fixed policy, which generates the target sentence only from the source sentence that is delayed by `k` tokens. However, their model for simultaneous translation relies only on the source sentence. In this research, we concentrate on the `wait-k` approach with fixed policy, so that the amount of input textual context can be controlled to analyze better whether multimodality is effective in SNMT.

Multimodal NMT (MNMT) for full-sentence machine translation has been developed to enrich text modality by using visual information (Hitschler et al., 2016; Specia et al., 2016; Elliott and Kádár, 2017). While the improvement brought by visual features is moderate, their usefulness is proven by Caglayan et al. (2019). They showed that MNMT models are able to capture visual clues under limited textual context, where source sentences are synthetically degraded by color deprivation, entity masking, and progressive masking. However, they use an artificial setting where they deliberately deprive the models of source-side textual context by masking. However, our research has discovered an actual end-task and has shown the effectiveness of using multimodal data for it. Compared with the entity masking experiments (Caglayan et al., 2019), where they use a model exposed to only `k` words, our model starts by waiting for the first `k` source words and then generates each target word after receiving every new source token, eventually seeing all input text.

In MNMT, visual features are incorporated into standard machine translation in many ways. Doubly-attentive models are used to capture the textual and visual context vectors independently and then combine these context vectors in a concatenation manner (Calixto et al., 2017) or hierarchical manner (Libovický and Helcl, 2017). Some studies use visual features in a multitask learning scenario (Elliott and Kádár, 2017; Zhou et al., 2018). Also, recent work on MNMT has partly addressed lexical ambiguity by using visual information (Elliott et al., 2017; Lala and Specia, 2018; Gella et al., 2019) showing that using textual context with visual features outperform unimodal models.

In our study, visual features are extracted using image processing techniques and then integrated into an SNMT model as additional information, which is supposed to be useful to predict missing words in a simultaneous translation scenario. To the best of our knowledge, this is the first work that incorporates external knowledge into an SNMT model.

3 Multimodal Simultaneous Neural Machine Translation Architecture

Our main goal is to investigate if image information would bring improvement on SNMT. As a result, two tasks could benefit from each other by combining them.

In this section, we describe our MSNMT model, which is composed by combining an SNMT framework `wait-k` (Ma et al., 2019) and a multimodal model (Libovický and Helcl, 2017). We base our model on the RNN architecture, which is widely used in MNMT research (Libovický and Helcl, 2017; Caglayan et al., 2017a; Elliott and Kádár, 2017; Zhou et al., 2018; Hirasawa et al., 2019). The model takes a sentence and its corresponding image as inputs. The decoder of the MSNMT model outputs the target language sentence in a simultaneous and multimodal manner by attaching

attention not only to the source sentence but also to the image related to the source sentence.[3]

3.1 Simultaneous Translation

We first briefly review standard NMT to set up the notations. The encoder of standard NMT model always takes the whole input sequence $\mathbf{X} = (x_1, ..., x_n)$ of length n where each x_i is a word embedding and produces source hidden states $\mathbf{H} = (h_1, ..., h_n)$. The decoder predicts the next output token y_t using $\mathbf{H}$ and previously generated tokens, denoted $\mathbf{Y}_{<t} = (y_1, ..., y_{t-1})$. The final output is calculated using the following equation:

$$p(\mathbf{Y}|\mathbf{X}) = \prod_{t=1}^{|\mathbf{Y}|} p(y_t|\mathbf{X}, y_{<t}) \quad (1)$$

Different from standard neural translation, in which each y_i is predicted using the entire source sentence $\mathbf{X}$, the simultaneous translation requires the model to translate concurrently with the growing source sentence. We incorporate the `wait-k` approach (Ma et al., 2019) for our simultaneous translation model. Instead of waiting for the whole sentence before translating, this model waits for only the first `k` tokens and starts to generate each target tokens after taking every new source token one by one. It stops taking new input tokens once the whole input sentence is on board. For example, if $\mathtt{k} = 3$, the first target token is predicted using the first 3 source tokens, and the second target token using the first 4 source tokens. The `wait-k` decoding probability $p_{\texttt{wait-k}}$ is:

$$p_{\texttt{wait-k}}(\mathbf{Y}|\mathbf{X}) = \prod_{t=1}^{|\mathbf{Y}|} p(y_t|\mathbf{X}_{\leq g(t)}, y_{<t}) \quad (2)$$

where $g(t)$ is the `wait-k` policy function which decides how much input text to read and translate, $\mathbf{X}_{\leq g(t)} = (x_1, ..., x_{g(t)})$ and $g(t)$ is $0 \leq t \leq n$. $g(t)$ is defined as follows:

$$g(t) = \min\{k + t - 1, n\} \quad (3)$$

When $k + t - 1$ is over source length n, $g(t)$ is fixed to n, which means the remaining target tokens (including current step) are generated using the full source sentence. For full sentence translation, $g(t)$ is constant $g(t) = n$.

[3]Our code is publicly available at: https://github.com/toshohirasawa/mst. We fixed our code based on the comments of Ozan Caglayan.

3.2 Multimodal Translation

We use a hierarchical attention combination technique (Libovický and Helcl, 2017) to incorporate visual and textual features into an MNMT model. This model calculates the independent context vectors from the textual features $\boldsymbol{h}^{\text{txt}} = (h_1^{\text{txt}}, ..., h_n^{\text{txt}})$ and the visual features $\boldsymbol{h}^{\text{img}} = (h_1^{\text{img}}, ..., h_m^{\text{img}})$, which are extracted by the textual encoder and the image processing model, respectively. It then combines the resulting two vectors using a second attention mechanism, which helps to perform simultaneous translation taking into account visual information.

Specifically, we compute the context vectors c_i^{f} for each image ($\text{f} = \text{img}$) and text ($\text{f} = \text{txt}$) modality independently using the following equations:

$$e_{i,j}^{\text{f}} = \Omega^{\text{f}}(s_i, h_j^{\text{f}}) \quad (4)$$

$$\alpha_{i,j}^{\text{f}} = \frac{\exp(e_{i,j}^{\text{f}})}{\sum_{l=1}^{|\mathbf{h}^{\text{f}}|} \exp(e_{i,l}^{\text{f}})} \quad (5)$$

$$c_i^{\text{f}} = \sum_{j=1}^{|\mathbf{h}^{\text{f}}|} \alpha_{i,j}^{\text{f}} h_j^{\text{f}} \quad (6)$$

where Ω^{f} is a feedforward network for each modality f; s_i is i-th decoder hidden state.

We project these image and text context vectors into a common space and compute another distribution over the projected context vectors and their corresponding weighted average using the second attention:

$$\tilde{e}_i^{\text{f}} = \Psi(s_i, c_i^{\text{f}}) \quad (7)$$

$$\beta_i^{\text{f}} = \frac{\exp(\tilde{e}_i^{\text{f}})}{\sum_{\text{r}\in\{\text{img,txt}\}} \exp(\tilde{e}_i^{\text{r}})} \quad (8)$$

$$\tilde{c}_i = \sum_{\text{r}\in\{\text{img,txt}\}} \beta_i^{\text{r}} W^{\text{r}} c_i^{\text{r}} \quad (9)$$

where Ψ is a feedforward network. Equation 8 calculates the second attention to combine the image and text vectors. W^{r} is a weight matrix used to compute the context vector $\tilde{c}_i$ calculated from image and text features. The final hypothesis $\mathbf{Y}$ has the probability:

$$p_{\text{mnmt}}(\mathbf{Y}|\mathbf{X}, \mathbf{Z}) = \prod_{t=1}^{|\mathbf{Y}|} p(y_t|\mathbf{X}, \mathbf{Z}, y_{<t}) \quad (10)$$

where $\mathbf{Z}$ represents input image features.

3.3 Multimodal Simultaneous Neural Machine Translation

In this subsection, we describe the structure of the MSNMT model, which is a combination of the models described in Sections 3.1 and 3.2. The method for calculating the image context vector is the same as for MNMT; however, the text context vector (Equation 6) for the t-th step is calculated as follows:

$$\hat{c}_i^{\text{txt}} = \sum_{j=1}^{g(t)} \alpha_{i,j}^{\text{txt}} h_j^{\text{txt}} \tag{11}$$

Thus $\hat{c}_i^{\text{txt}}$ is calculated from the input text prefix determined by `wait-k` policy function $g(t)$. Then we apply the second attention to $\hat{c}_i^{\text{txt}}$ and c_i^{img} in order to calculate $\tilde{c}_i$ (Equation 9).

The decoding probability becomes as follows:

$$p_{\text{msnmt}}(\mathbf{Y}|\mathbf{X},\mathbf{Z}) = \prod_{t=1}^{|\mathbf{Y}|} p(y_t|\mathbf{X}_{\leq g(t)},\mathbf{Z},y_{<t}) \tag{12}$$

4 Experimental Setup

4.1 Dataset

We experiment with our model in four translation directions consisting of 5 languages: English (En), German (De), French (Fr), Czech (Cs), and Japanese (Ja). All language pairs include En on the source side.

We used the train, development, and test sets from the Multi30k (Elliott et al., 2016) dataset published in the WMT16 Shared Task, which is a benchmark dataset generally used in MNMT research (Libovický and Helcl, 2017; Caglayan et al., 2019; Elliott and Kádár, 2017; Zhou et al., 2018; Hirasawa et al., 2019) for En→De, En→Fr and En→Cs.

Nakayama et al. (2020) released F30kEnt-JP dataset[4] which contains Japanese translations of first two original English captions for each image of the Flickr30k Entities dataset (Plummer et al., 2017). They follow the same annotation rules as the Flickr30k Entities dataset using exactly the same tags with entity types and IDs. We preprocessed this data as follows: 1) The parallel En→Ja data was created by taking alignment using corresponding IDs assigned to each Japanese translation entity with the IDs of Flickr30k entities.[5] 2) The created parallel data was aligned with its corresponding images using text files named $(image\_id).txt$ corresponding to each image in Flickr30k. 3) Finally, the created multimodal data was split to train, dev, and test following data splits of Multi30k using the same Multi30k image IDs. Note that the English side of En→Ja parallel data extracted from F30kEnt-JP and English side of Multi30k data are thought to be somewhat comparable but not strictly the same while their corresponding images are the same.

Data split for all language pairs were as follows: training set, 29,000 sentence pairs, development set, 1,014 sentence pairs, and 1,000 sentence pairs for the test set. This dataset's average sentence length is 12-13 tokens for En, De, Fr, Cs and 20 tokens for Ja.

We limit the vocabulary size of the source and the target languages after concatenating them to 10,000 sub-words (Sennrich et al., 2016). All sentences are preprocessed with lower-casing, tokenizing, and normalizing the punctuation using the Moses script[6]. To tokenize Japanese sentences, we used MeCab[7] with the IPA dictionary.

Visual features are extracted using pre-trained ResNet (He et al., 2016). Technically, we encode all images in Multi30k with ResNet-50 and pick out the hidden state in the pool5 layer as a 2,048-dimension visual feature.

4.2 Systems

We compare the following models: **1. SNMT:** We use only text modality for training data as a baseline for each `wait-k` model. **2. MSNMT:** We use image modality along with text modality for a training data for each `wait-k` model.

To train the above models, we utilize attention NMT (Bahdanau et al., 2015) with a 2-layer unidirectional GRU encoder and a 2-layer conditional GRU decoder. We use the open-source implementation of the `nmtpytorch` toolkit v3.0.0 (Caglayan et al., 2017b). We first pre-train the MSNMT model for each `k` until convergence using only text data and use zeros for visual features. Then we continue training MSNMT on multimodal data for

[4] https://github.com/nlab-mpg/Flickr30kEnt-JP

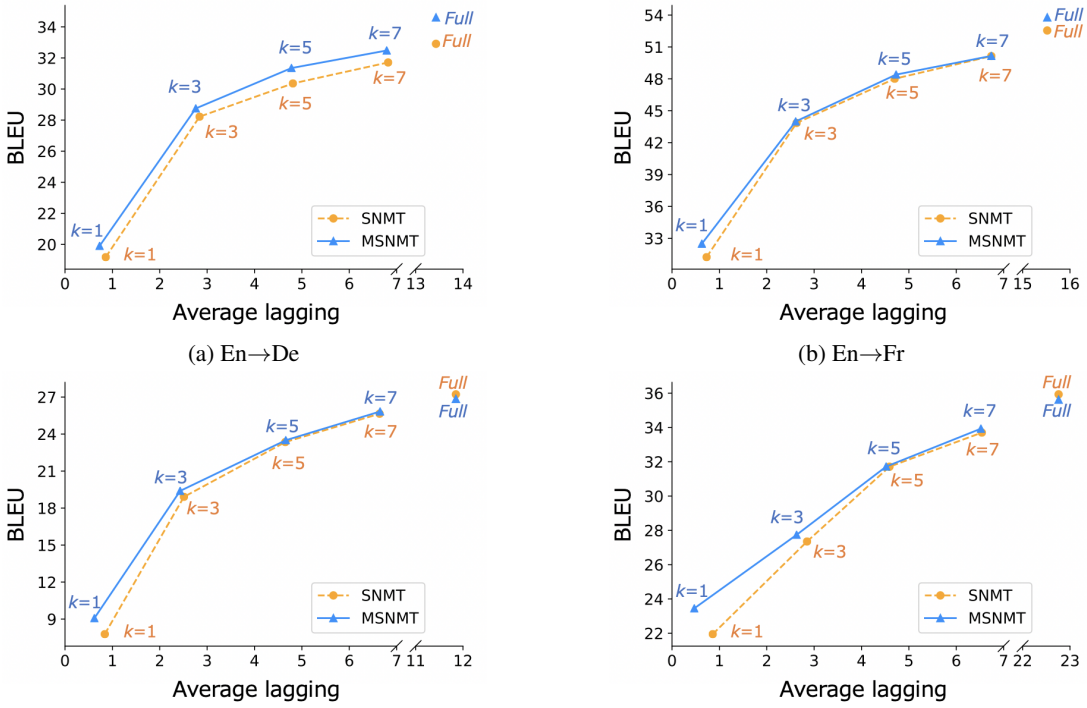
[5] We used the second translations due to some empty translations of the first captions.

[6] We applied preprocessing using task1-tokenize.sh from https://github.com/multi30k/dataset.

[7] http://taku910.github.io/mecab, version 0.996.

wait-k	En→De		En→Fr		En→Cs		En→Ja	
	S	M	S	M	S	M	S	M
1	19.18	†**19.90**	31.23	†**32.49**	7.78	†**9.07**	21.95	†**23.45**
3	28.22	†**28.75**	43.85	**43.99**	18.91	†**19.39**	27.35	†**27.74**
5	30.38	†**31.48**	48.01	†**48.40**	23.35	**23.50**	31.71	**31.72**
7	31.72	**32.14**	50.14	**50.16**	25.65	**25.83**	33.70	**33.93**
Full	34.64	**34.84**	53.55	**53.78**	**27.22**	26.85	**35.93**	35.62

Table 1: BLEU scores of SNMT (S) and MSNMT (M) models for four translation directions on test set. Results are the average of four runs. **Bold** indicates the best BLEU score for each `wait-k` for each translation direction. "†" indicates statistical significance of the improvement over SNMT.



(a) En→De (b) En→Fr

(c) En→Cs (d) En→Ja

Figure 1: Average Lagging scores. Results are the average of four runs.

each `k`. We employ early-stopping: the training was stopped when the BLEU score did not increase on the development set for 10 epochs for MSNMT pre-training, 5 epochs for MSNMT fine-tuning, and 15 epochs for SNMT training.

In order to keep our experiments as pure as possible, we will not use additional data or other types of models. It will allow us to control the amount of input textual context, so we can easily analyze the relationship between the amount of textual and visual information.

4.3 Hyperparameters

We use the same hyperparameters for SNMT and MSNMT for a fair comparison as follows. All models have word embeddings of 200 and recurrent layers of dimensionality 400 units with 2way sharing of embeddings in the network. We used Adam (Kingma and Ba, 2015) with a learning rate of 0.0004. Decoders were initialized with zeros. We used a minibatch size of 64 for training and 32 for fine-tuning. Rates of dropout applied on source embeddings, source encoder states and pre-softmax activations were 0.4, 0.5, and 0.5, respectively. We set the max length of the input to 100. `wait-k` experiments were conducted for 1, 3, 5, 7, and Full settings. For MSNMT only hyperparameters, the sampler type was set to approximate, and channels were set to 2048. The fusion type was set to hierarchical mode.

4.4 Evaluation

We report BLEU scores calculated using Moses' multi-bleu.perl, which is a widely used evalu-

wait-k	En→De		En→Fr		En→Cs		En→Ja	
	C	I	C	I	C	I	C	I
1	†**19.90**	8.19	†**32.49**	18.00	†**9.07**	6.83	†**23.45**	17.57
3	†**28.75**	26.78	†**43.99**	42.31	†**19.39**	18.78	†**27.74**	24.51
5	**31.48**	31.08	**48.40**	48.19	†**23.50**	22.81	†**31.72**	28.57
7	†**32.14**	32.04	**50.16**	50.15	†**25.83**	25.09	†**33.93**	31.03
Full	**34.84**	34.40	†**53.78**	53.10	**26.85**	26.84	**35.62**	35.59

Table 2: Image Awareness results on the test set. BLEU scores of MSNMT Congruent (C) and Incongruent (I) settings for four translation directions. Results are the average of four runs. **Bold** indicates the best BLEU score for each wait-k for each translation direction. "†" indicates the statistical significance of the improvement over Incongruent settings.

ation metric in MT, on our test sets for each wait-k model.[8] Statistical significance ($p < 0.05$) on the difference of BLEU scores was tested by Moses' *bootstrap-hypothesis-difference-significance.pl*. "Full" means that the whole input sentence is used as an input for the model to start translating. All reported results are the average of four runs using four different random seeds.

Additionaly, we use open-sourced Average Lagging (AL) latency metric proposed by Ma et al. (2019) to evaluate the latency for SNMT and MSNMT systems.[9] It calculates the degree of out of sync time with the input, in terms of the number of source tokens as follows:

$$\mathrm{AL}_g(\mathbf{X}, \mathbf{Y}) = \frac{1}{\tau_g(|\mathbf{X}|)} \sum_{t=1}^{\tau_g(|\mathbf{X}|)} g(t) - \frac{t-1}{r} \quad (13)$$

where $r = |\mathbf{Y}|/|\mathbf{X}|$ is the target-to-source length ratio and τ_g is the decoding step when source sentence finishes:

$$\tau_g(|\mathbf{X}|) = \min\{t | g(t) = |\mathbf{X}|\} \quad (14)$$

5 Results

Table 1 illustrates the BLEU scores of MSNMT and SNMT models on the test set. MSNMT systems show significant improvements over SNMT systems for all language pairs when input textual information is limited. Note that the difference of BLEU scores between MSNMT and SNMT becomes larger as the k gets smaller, especially when the target language is distant from English in terms of word order (e.g. Cs and Ja). On the other hand, the availability of more tokens during the decoding process ($k \geq 5$) leads to the text information becoming sufficient in some cases.

[8]Due to space constraints, we show results only for test sets.

[9]https://github.com/SimulTrans-demo/STACL

Figure 1 shows translation quality against AL for four language directions. In all these figures, we observe that, as k increases, the gap between BLEU scores for MSNMT and SNMT decreases. We also observe that AL scores are better for MSNMT as k decreases. From these results, it can be seen that in terms of latency, the smaller k is, the more beneficial the visual clues become.

6 Analysis

In this section, we provide a thorough analysis to further investigate the effect of visual data to produce a simultaneous translation by (a) providing adversarial evaluation; and (b) analyzing the impact of different word order for En→Ja language pair.

6.1 Adversarial Evaluation

In order to determine whether MSNMT systems are aware of the visual context (Elliott, 2018), we perform the adversarial evaluation on the test set. We present our system with correct visual data with its source sentence (Congruent) as opposed to random visual data as an input (Incongruent) (Elliott, 2018).Therefore, we reversed the order of 1,000 images of the test set, so there will be no overlapping congruent visual data. Then we reconstruct image features for those images to use as an input.

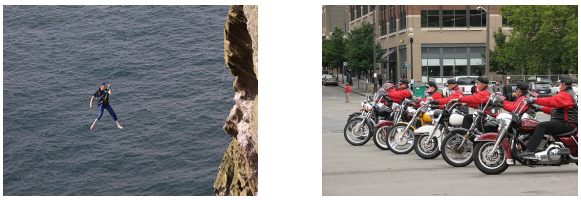
Results of image awareness experiments are shown in Table 2. We can see the large difference in BLEU scores between MSNMT congruent (C columns) and incongruent (I columns) settings when k are small. This implies that our proposed model utilizes images for translation by learning to extract needed information from visual clues. The interesting part is for a full translation, where scores for the incongruent setting are very close to those of the congruent setting. The reason is that when textual information is enough, visual information becomes not that relevant in some cases.

6.2 How Source-Target Word Order Affects Translation

In wait-k translations, for the En→Ja language pair with different word orders (SVO vs. SOV), some source tokens should be translated before they are presented to the decoder for grammaticality and fluency purposes. Hence, the model also needs to handle such cases well apart from the "usual" order. We hypothesized that MSNMT models, given additional visual information, are able to translate such cases better than SNMT models. Therefore, we investigated how many tokens were correctly translated that are not given as input yet.

First, we quantitatively analyze how well we can translate entities that are not presented from the source yet but should exist in target sentences. To align the source and target entities, we use the entities' annotation attached to both the source and target sentences. Given that annotated entities have the same IDs and tags for both English and Japanese, we can align, calculate, and extract those entities from source and target sentences. If the index of the first token of the aligned target entity is not given as input at timestep k yet, we count them for each k scenario as # total entities (Table 4). For example, in Table 3 a wait-3 model should start translating after a token "rappelling" is presented to the model. And if an ID of the entity of "海 (a body of water)" is in the target sentences but not in the inputted part yet, we count it as an entity that should be translated before being inputted to the model. Similarly, an entity of "断崖 (cliff)" is already presented to the model at timestep 5, so we do not count those entities. If the same entity ID appears more than once in one sentence, we exclude those entities due to the impossibility of alignments. Finally, for each model during decoding, if those entities are included in the model's translation results with a perfect match from pre-calculated # total entities, we consider them as correctly translated.[10]

Table 4 demonstrates the results. k column is to determine how many tokens a model waits before starting translating. Note that k=Full is not included because all entities are given at the time of translation. The reason that the total number of entities that were not inputted yet decreases when k increases (# total entities column) is that more entities are already available for the model for translation. wait-k columns show how many entities were correctly translated by wait-k SNMT and MSNMT models from # total entities for each k scenario. Columns Full show upper-bounds of how many entities can be correctly translated if the models were trained with full sentences for entities from each k. Comparing Full results to wait-k for both SNMT and MSNMT shows that it is hard to correctly translate entities when k is small. Furthermore, comparing wait-k results of SNMT to MSNMT, it can be seen that the smaller value of k, the better MSNMT can handle different source-target word order than SNMT.



(a) A person rappelling a cliff.

(b) Eight men on motorcycles.

Figure 2: Images presented in translation examples (Table 5).

As an example, we sampled sentences and their images from the En→Ja test set (Figure 2) to compare the outputs of our systems. Table 5 lists their translations generated by SNMT (S) and MSNMT (M) models. In the first example, an SNMT model with wait-3 could not predict "海 (sea, a body of water)" which appears at the end of the source sentence and generated an erroneous "岩 (rock)" which is not present neither in source text nor in a corresponding image. Contrarily, the MSNMT model with wait-3 was able to correctly predict "海 (body of water)" even before it was inputted by capturing visual information. When a full sentence is given as an input, MSNMT translated it correctly using more information, unlike SNMT, which translated only from the given text and generated incorrect "登って (climbing)" instead of "降りて (rappelling)". Interestingly, in the second example, the MSNMT model with wait-3 predicted "自転車 (bicycles)" instead of "オートバイ (motorcycles)" at the beginning of the sentence, while the SNMT model with wait-3 was not able to generate any vehicle entities. Also, both MSNMT models with wait-3 and Full correctly captured that there were eight men, whilst both SNMT models incorrectly predicted about one and two men. From these results, we can conclude that visual clues pos-

[10] We can not create # total entities from decoded tokens directly due to unavailability of entity annotations.

t	1	2	3	4	5	6	7	8	9	10	11											
Source	a	person	rappelling	a	cliff	above	a	body	of	water	.											
Target, k=3				海	の	上	に	ある	断崖	を	降り	て	いる	一	人	の	男性	。				
Entity count				✓					✗								✗					

Table 3: Example of En→Ja translation to count entities that should be translated before introducing it to a model in case of `wait-3` (see Figure 2a). A `wait-k` model starts translating after `k` tokens are inputted. Colors represent the same entities. ✓ indicates entities that are not presented to the model at timestep t yet and ✗ indicates entities that are already seen by the model at timestep t. We count only those entities marked with ✓ for # total entities (Table 4).

k	# total entities	# correct entities by S		# correct entities by M	
		`wait-k`	`Full`	`wait-k`	`Full`
1	1,343	251	**716**	**270**	707
3	852	229	**433**	**242**	432
5	502	147	**247**	**151**	243
7	320	106	**160**	106	159

Table 4: Number of entities that were correctly translated before being presented to the model by SNMT (S) and MSNMT (M) models with their for each `k`. Results are the average of four runs.

Source	a person rappelling a cliff above a body of water .
Target	海の上にある断崖を降りている一人の男性。
S `wait-3`	誰かが、岩の上で崖を登る。 (someone climbs a cliff on a rock.)
M `wait-3`	人が海の上で崖を降りている。 (a person is rappelling a cliff above the sea.)
S `Full`	人が水域の上の崖を登っている。 (a person is climbing a cliff above a body of water.)
M `Full`	人が水域の上で崖を降りている。 (a person is rappelling a cliff above a body of water.)
Source	eight men on motorcycles dressed in red and black are all lined up on the side of the street .
Target	赤と黒の服を着たオートバイに乗っている８人の男性が通りの脇にずらりと並んでいる。
S `wait-3`	白い服を着て、黒と黒の服を着た１人の男性が、通りの脇に並んでいる。 (a man in white and black and black is standing beside the street.)
M `wait-3`	自転車に乗っている赤と黒の服を着た８人の男性が、通りの側面にある。 (eight men in red and black clothes riding a bicycle are on the side of the street.)
S `Full`	赤と黒の服を着た、オートバイに乗った２人の男性が、通りの脇で並んでいる。 (two men on motorcycles, dressed in red and black, line up by the side of the street.)
M `Full`	赤と黒の服を着た、オートバイに乗った８人の男性が、通りの側面に並んでいる。 (eight men on motorcycles, dressed in red and black, line the side of the street.)

Table 5: Examples of En→Ja translations from test set using SNMT (S) and MSNMT (M) models (also refer to Figure 2). In () are shown their English meanings. The same colors indicate the same entity types.

itively impact generated translations where there is still a lack of textual information, especially when we deal with language pairs with different word order.

7 Conclusion

In this paper, we proposed a multimodal simultaneous neural machine translation approach, which takes advantage of visual information as an additional modality to compensate for the shortage of input text information in the simultaneous neural machine translation. We showed that in a `wait-k` setting, our model significantly outperformed its text-only counterpart in situations where only a few input tokens are available to begin translation. We showed the importance of the visual information for simultaneous translation, especially in the low latency setup and for a language pair with word-order differences. We hope that our proposed method can be explored even further for various tasks and datasets.

In this paper, we created a separate model for each value of `wait-k`. However, in future work, we plan to experiment on having a single model for all `k` values (Zheng et al., 2019b). Furthermore, we acknowledge the importance of investigating MSNMT effects on more realistic data (e.g. TED), where the utterance does not necessarily match a shown image while speaking and/or where its context can not be guessed from the shown image.

Acknowledgments

We are immensely grateful to Raj Dabre and Rob van der Goot who provided expertise, support, and insightful comments that significantly improved the manuscript. We would also like to show our gratitude to Desmond Elliot for valuable feedback and paper discussions. We want to thank Ozan Caglayan for pointing out critical bugs in our previous implementation.

References

Naveen Arivazhagan, Colin Cherry, Wolfgang Macherey, Chung-Cheng Chiu, Semih Yavuz, Ruoming Pang, Wei Li, and Colin Raffel. 2019. Monotonic infinite lookback attention for simultaneous machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1313–1323.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the 3rd International Conference on Learning Representations*.

Ozan Caglayan, Walid Aransa, Adrien Bardet, Mercedes García-Martínez, Fethi Bougares, Loïc Barrault, Marc Masana, Luis Herranz, and Joost van de Weijer. 2017a. LIUM-CVC submissions for WMT17 multimodal translation task. In *Proceedings of the Second Conference on Machine Translation*, pages 432–439.

Ozan Caglayan, Mercedes García-Martínez, Adrien Bardet, Walid Aransa, Fethi Bougares, and Loïc Barrault. 2017b. NMTPY: A flexible toolkit for advanced neural machine translation systems. *The Prague Bulletin of Mathematical Linguistics*, 109(1):15–28.

Ozan Caglayan, Pranava Madhyastha, Lucia Specia, and Loïc Barrault. 2019. Probing the need for visual context in multimodal machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4159–4170.

Iacer Calixto, Qun Liu, and Nick Campbell. 2017. Doubly-attentive decoder for multi-modal neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1913–1924.

Fahim Dalvi, Nadir Durrani, Hassan Sajjad, and Stephan Vogel. 2018. Incremental decoding and training methods for simultaneous translation in neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 493–499.

Desmond Elliott. 2018. Adversarial evaluation of multimodal machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2974–2978.

Desmond Elliott, Stella Frank, Loïc Barrault, Fethi Bougares, and Lucia Specia. 2017. Findings of the second shared task on multimodal machine translation and multilingual image description. In *Proceedings of the Second Conference on Machine Translation*, pages 215–233.

Desmond Elliott, Stella Frank, Khalil Sima'an, and Lucia Specia. 2016. Multi30k: Multilingual English-German image descriptions. In *Proceedings of the 5th Workshop on Vision and Language*, pages 70–74.

Desmond Elliott and Àkos Kádár. 2017. Imagination improves multimodal translation. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 130–141.

Spandana Gella, Desmond Elliott, and Frank Keller. 2019. Cross-lingual visual verb sense disambiguation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1998–2004.

Alvin Grissom II, He He, Jordan Boyd-Graber, John Morgan, and Hal Daumé III. 2014. Don't until the final verb wait: Reinforcement learning for simultaneous machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1342–1352.

Jiatao Gu, Graham Neubig, Kyunghyun Cho, and Victor OK Li. 2017. Learning to translate in real-time with neural machine translation. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1, Long Papers)*, pages 1053–1062.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778.

Tosho Hirasawa, Hayahide Yamagishi, Yukio Matsumura, and Mamoru Komachi. 2019. Multimodal machine translation with embedding prediction. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 86–91.

Julian Hitschler, Shigehiko Schamoni, and Stefan Riezler. 2016. Multimodal pivots for image caption translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2399–2409.

Diederik Kingma and Jimmy Ba. 2015. Adam: a method for stochastic optimization. In *The International Conference on Learning Representations*.

Chiraag Lala and Lucia Specia. 2018. Multimodal lexical translation. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, pages 3810–3817.

Jindřich Libovický and Jindřich Helcl. 2017. Attention strategies for multi-source sequence-to-sequence learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 196–202.

Mingbo Ma, Liang Huang, Hao Xiong, Renjie Zheng, Kaibo Liu, Baigong Zheng, Chuanqiang Zhang, Zhongjun He, Hairong Liu, Xing Li, Hua Wu, and Haifeng Wang. 2019. STACL: Simultaneous translation with implicit anticipation and controllable latency using prefix-to-prefix framework. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3025–3036.

Shigeki Matsubara, Kiyoshi Iwashima, Nobuo Kawaguchi, Katsuhiko Toyama, and Yoichi Inagaki. 2000. Simultaneous Japanese-English interpretation based on early prediction of English verb. In *Proceedings of The Fourth Symposium on Natural Language Processing*, pages 268–273.

Hideki Nakayama, Akihiro Tamura, and Takashi Ninomiya. 2020. A visually-grounded parallel corpus with phrase-to-region linking. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 4204–4210.

Yusuke Oda, Graham Neubig, Sakriani Sakti, Tomoki Toda, and Satoshi Nakamura. 2015. Syntax-based simultaneous translation through prediction of unseen syntactic constituents. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 198–207.

Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. 2017. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. *International Journal of Computer Vision*, 123(1):74–93.

Kilian G Seeber. 2015. Simultaneous interpreting. In *The Routledge Handbook of Interpreting*, pages 91–107.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistic*, pages 1715–1725.

Lucia Specia, Stella Frank, Khalil Sima'an, and Desmond Elliott. 2016. A shared task on multimodal machine translation and crosslingual image description. In *Proceedings of the First Conference on Machine Translation: (Volume 2: Shared Task Papers)*, pages 543–553.

Baigong Zheng, Renjie Zheng, Mingbo Ma, and Liang Huang. 2019a. Simpler and faster learning of adaptive policies for simultaneous translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1349–1354.

Baigong Zheng, Renjie Zheng, Mingbo Ma, and Liang Huang. 2019b. Simultaneous translation with flexible policy via restricted imitation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5816–5822.

Renjie Zheng, Mingbo Ma, Baigong Zheng, and Liang Huang. 2019c. Speculative beam search for simultaneous translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1395–1402.

Renjie Zheng, Mingbo Ma, Baigong Zheng, Kaibo Liu, and Liang Huang. 2020. Opportunistic decoding with timely correction for simultaneous translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 437–442.

Mingyang Zhou, Runxiang Cheng, Yong Jae Lee, and Zhou Yu. 2018. A visual attention grounding neural model for multimodal machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3643–3653.

Diving Deep into Context-Aware Neural Machine Translation

Jingjing Huo[1,2] **Christian Herold**[2] **Yingbo Gao**[2] **Leonard Dahlmann**[1]
Shahram Khadivi[1] **Hermann Ney**[2]
[1]eBay, Inc., Aachen, Germany
{jihuo, fdahlmann, skhadivi}@ebay.com
[2]Human Language Technology and Pattern Recognition Group
RWTH Aachen University, Aachen, Germany
{surname}@i6.informatik.rwth-aachen.de

Abstract

Context-aware neural machine translation (NMT) is a promising direction to improve the translation quality by making use of the additional context, e.g., document-level translation, or having meta-information. Although there exist various architectures and analyses, the effectiveness of different context-aware NMT models is not well explored yet. This paper analyzes the performance of document-level NMT models on four diverse domains with a varied amount of parallel document-level bilingual data. We conduct a comprehensive set of experiments to investigate the impact of document-level NMT. We find that there is no single best approach to document-level NMT, but rather that different architectures come out on top on different tasks. Looking at task-specific problems, such as pronoun resolution or headline translation, we find improvements in the context-aware systems, even in cases where the corpus-level metrics like BLEU show no significant improvement. We also show that document-level back-translation significantly helps to compensate for the lack of document-level bi-texts.

1 Introduction

Even though machine translation (MT) has greatly improved with the emergence of neural machine translation (NMT) (Sutskever et al., 2014; Bahdanau et al., 2015) and more recently the Transformer architecture (Vaswani et al., 2017), there remain challenges which can not be solved by using sentence-level NMT systems. Among other issues, this includes the problem of inter-sentential anaphora resolution (Guillou et al., 2018) or the consistent translation across a document (Läubli et al., 2018), for which the system inevitably needs document-level context information.

In recent years, many works have focused on changing existing NMT architectures to incorporate context information in the translation process (Tiedemann and Scherrer, 2017; Bawden et al., 2018; Voita et al., 2018). However, often times results are reported only on very specific tasks (most commonly subtitle translation), making it difficult to assess the potential of the different methods in a more general setting. This, together with the fact that big improvements are typically reported on low resource tasks, gives the impression that document-level NMT mostly improves due to regularization rather than from leveraging the additional context information. In this work we want to give a more complete overview of the current state of document-level NMT by comparing various approaches on a variety of different tasks including an application-oriented E-commerce setting. We discuss both, widely used performance metrics, as well as highly task-specific observations.

Another important aspect when talking about document-level NMT is the applicability in "real life" settings. There, when faced with a low resource data scenario, back-translation is an established way of greatly improving system performance (Sennrich et al., 2016a). However, to the best of our knowledge, the effect of back-translation data obtained and used by context-aware models has never been explored before. The main contributions of this paper are summarized below:

- We explore several existing context-aware architectures on four diverse machine translation tasks, consisting of different domains and data quantities.
- We examine the usage of context aware embeddings created by pre-trained monolingual models and study to what extent these embeddings can be simplified.
- We conduct corpus studies and extensive analysis on corpus specific phenomena like pronoun resolution or headline translation to give

Proceedings of the 5th Conference on Machine Translation (WMT), pages 604–616
Online, November 19–20, 2020.

an interpretation of the potential improvements from leveraging context information.

- We study the effects of utilizing document-level monolingual data via back-translation and report significant improvements particularly for document-level NMT systems.

2 Related Works

The discourse- or document-level translation is a long-standing and unsolved topic in the machine translation community (Mitkov, 1999; Carpuat, 2009; Hardmeier, 2014). Although neural machine translation (Sutskever et al., 2014; Bahdanau et al., 2015; Vaswani et al., 2017) has recently become the dominant translation paradigm that provides superior performance, the independence between sentences is still the fundamental assumption taken for granted by most NMT systems. This means, that discourse-level phenomena between sentences such as pronominal reference, consistent lexical choice, and verbal tenses, etc. can not be addressed by these sentence-level NMT systems (Läubli et al., 2018; Guillou et al., 2018). The current NMT approaches tackling inter-sentential discourse phenomena can be roughly categorized into three aspects, augmenting NMT by

- adding source-side context
- including both source- and target-side context
- utilizing source- and/or target-side document-level monolingual data

To include the source-side context, Tiedemann and Scherrer (2017) concatenate consecutive sentences as input to the NMT system, while Jean et al. (2017); Bawden et al. (2018); Zhang et al. (2018) use an additional encoder to extract contextual information from a few previous source-side sentences. These works only consider a local context, including a few previous sentences. Some researches seek to capture the global document context; Wang et al. (2017) summarize the global context from all previous sentences in a document with a pre-trained hierarchical RNN and then use it for updating decoder states. Very recently, Chen et al. (2020) proposed a discourse structure-based encoder that takes account of the discourse structure information of the input document.

For adding additional target-side context, Tiedemann and Scherrer (2017); Agrawal et al. (2018) conduct multi-sentences decoding and observe only a minor improvement. Maruf and Haffari (2018) apply cache-based models to store vector representations for both source- and target-side context. Similarly, Tu et al. (2018) augment their NMT system with an external cache to memorize the translation history. Werlen et al. (2018) integrate two hierarchical attention networks (HAN) (Yang et al., 2016) in the NMT model to take account for source and target context. Maruf et al. (2019) apply a hierarchical attention module on sentences and words in the context to select contextual information that is more relevant to the current sentence.

For incorporating document-level monolingual data from the source language, Zhu et al. (2020) use BERT (Devlin et al., 2019) to model the source-side context and integrate it with the encoder and decoder of the NMT model. Junczys-Dowmunt (2019) share the parameters of a BERT-style encoder trained on monolingual documents with the MT model.

To utilize the document-level monolingual data from the target language, Junczys-Dowmunt (2019) also submit a system that trained on the combination of real and synthetic document-parallel data obtained by back-translation. However, they do not consider document-level back-translation. Voita et al. (2019a) proposed a document-level post-editing system which is trained only using the monolingual document-level corpus.

Recently, there has been a tendency in the community to conclude that the context used in a context-aware MT model works as regularisation or noise generator. Kim et al. (2019) compare several multi-encoders methods and claim that including this additional information can improve translation performance, but it is mostly due to the regularization effect rather than the contextual information. Li et al. (2020) also compare some context-aware architectures by replacing the real context with some random signal and show that random signals can achieve the same level improvement as the real context. However, it should be taken with a grain of salt since solving this task, along with the analysis, is quite challenging. There are many impact factors from the architecture, the data at hand, to the metric being used for evaluation.

One issue that can not be ignored in all discourse-related researches is the problem of evaluation. Since some discourse-level phenomena between sentences appear less frequently, although relevant, there is doubt if the metrics like BLEU score (Papineni et al., 2002) can capture these complex re-

lationships (Le Nagard and Koehn, 2010; Hardmeier and Federico, 2010). To get more insights into the capacities dealing with discourse-level phenomena of their MT models, some researchers use more targeted evaluation scores (Wong et al., 2020), like the Accuracy of Pronoun Translation (APT) Werlen and Popescu-Belis (2017), or they evaluate their systems on some specific test suites that contain more and more complex discourse phenomena (Hardmeier et al., 2015; Guillou et al., 2018; Müller et al., 2018; Voita et al., 2019b).

3 Document-level NMT

In this section, we first describe several commonly used context-aware NMT architectures and highlight the differences among them, largely following the work by Kim et al. (2019). Afterwards, we describe one radical attempt to represent the document-level context in one single embedding vector using BERT (Devlin et al., 2019). Finally, we explain our proposed paradigm to use document-level back-translation in detail. Note that in this work, we consistently use Transformer (Vaswani et al., 2017) as our basic architecture and modify it accordingly.

3.1 Context-Aware Architectures

Given a source sentence in a document to be translated, in order to exploit the source-side context from its previous sentences in the same document, a simple and straightforward technique is to concatenate these contextual sentences with the current source sentence (Tiedemann and Scherrer, 2017; Agrawal et al., 2018). Similarly, if the previous and current target sentences are to be generated together, i.e. $\underline{\mathrm{e}}_1^I = e_1^{I_{\mathrm{pre}}}$ BREAK $e_1^{I_{\mathrm{cur}}}$, then the target-side context can also be considered by the model (Tiedemann and Scherrer, 2017). Two additional special tokens are introduced to indicate the boundary between sentences and the beginning of a document, respectively. In this case, there is no modification of the model architecture itself, as seen in Figure 1.

An alternative way to model the source-side context is via an additional encoder, as shown in Figure 2. The previous sentence f_{pre} is fed into an additional encoder to obtain the hidden representation of the source context sentence $h_{j_{\mathrm{pre}}}^{L-1}$. At the last layer of the encoder, the source representation h_j^{L-1} attends to $h_{j_{\mathrm{pre}}}^{L-1}$ and outputs the combined hidden representation c_j^L (Voita et al., 2018). Then,

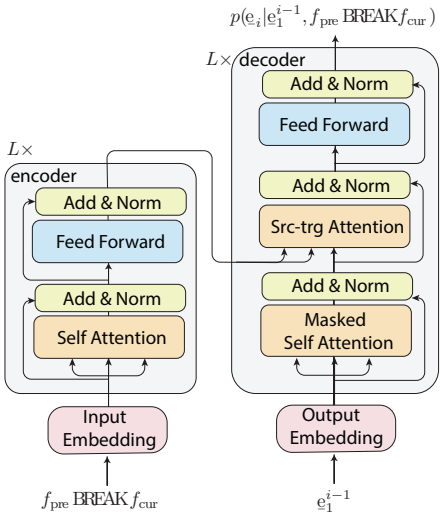


Figure 1: Single Encoder (2to2) approach only considering the one previous source sentence as context.

a gating mechanism (Bawden et al., 2018) between h_j^L and c_j^L is followed:

$$g_j = \sigma(W_g[h_j^L, c_j^L] + b_g) \quad (1)$$

$$o_j = g_i \odot W_s h_j^L + (1 - g_i) \odot W_c c_j^L \quad (2)$$

We refer to this approach as "Multi-Encoders (Out.)".

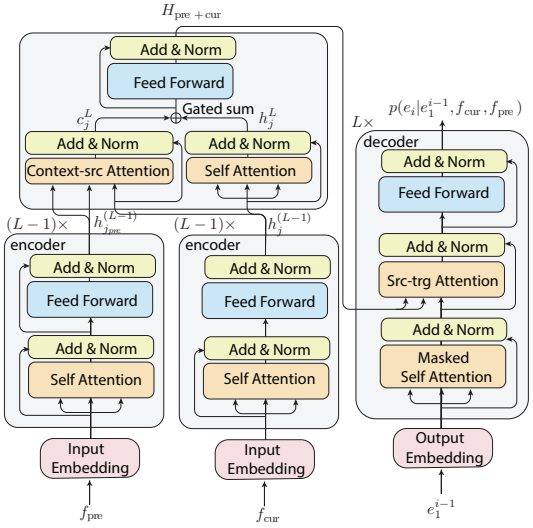


Figure 2: Multi-Encoders Out-side of decoder approach (Out.).

Another way to do the integration is to keep the representation of the current source sentence and the representation of the contexts separate and allow the decoder to have access to the context representations. Figure 3 shows a sequential integration inside of the decoder, where the decoder firstly attends to the current source representation,

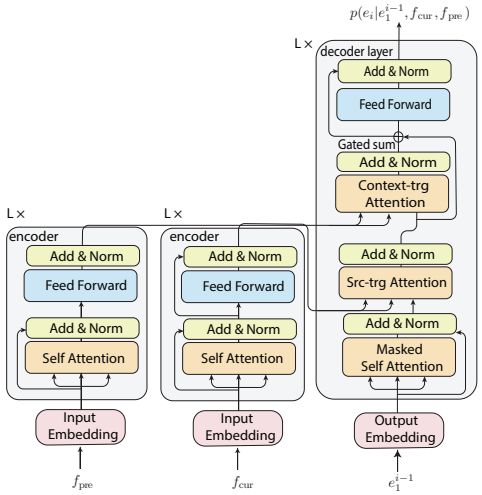


Figure 3: Multi-Encoders followed by attention components Inside of decoder Sequentially (In. Seq.).

then its output attends to the context representation (Zhang et al., 2018). The same gating mechanism as in the Multi-Encoders (Out.) approach is used between the two attention outputs. We refer to this approach that uses multi-encoders followed by attention components inside of decoder sequentially as "Multi-Encoders (In. Seq.)".

Figure 4 shows a parallel integration of the context inside of the decoder, where the decoder attends to the source and context representation in parallel and the outputs of them are combined again using a gating mechanism (Bawden et al., 2018). In this paper, we call this approach using multiple encoders followed by attention components inside the decoder in parallel "Multi-Encoders (In. Par.)".

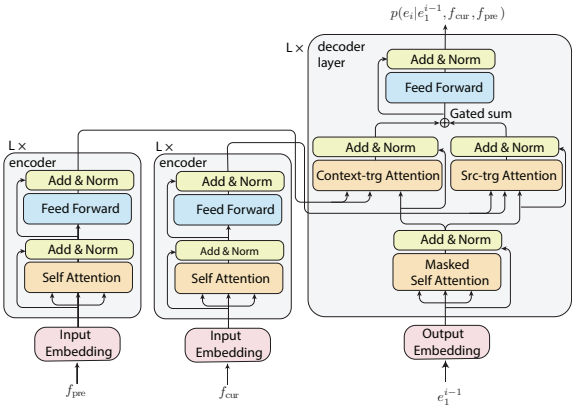


Figure 4: Multi-Encoders followed by attention components Inside of decoder in Parallel (In. Par.).

In addition, we use "WordEmb (In. Par.)" to refer to the approach that only uses word embeddings without any hidden layers to model the context and integrate it following the Multi-Encoders (In. Par.).

Considering that a pre-trained model like ELMo (Peters et al., 2018) or BERT (Devlin et al., 2019) can capture rich representations of the input, it is apparent that one can also use it to model contextual information. Figure 5 shows the BERT-fused model proposed in Zhu et al. (2020), which uses a BERT encoder to obtain the BERT hidden representations H_B on the concatenation of the context sentence f_{pre} and the current source sentence f_{cur}. H_B is further fused into each layer of the encoder

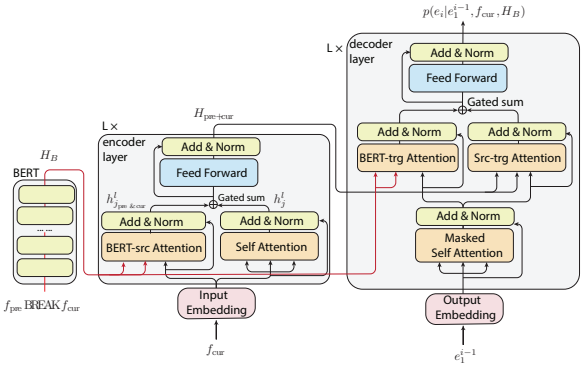


Figure 5: BERT sequence embeddings approach (Zhu et al., 2020).

and decoder of the NMT model using the attention mechanism to obtain the context representation. Instead of using the summation operation like in the original paper, we combine the context representation $h^l_{j_{\text{pre\&cur}}}$ and source representation h^l_j with a gating mechanism on the encoder side. Similar operation for the integration on the decoder side is used. This approach corresponds to the "BERT sequence embeddings (emb.)" approach in our main results in Table 3.

3.2 Single Embedding Vector as Context Representation

The introduction of additional encoders or attention components in the approaches mentioned in Section 3.1 brings a large number of parameters, which is not always ideal. Further, we propose one radical attempt to summarize the document-level context into one single embedding vector. We average the embeddings in the context representation H_B obtained by BERT to obtain one single mean-pooled embedding and then concatenate it with the word embeddings of the current source sentence along time axis (T-axis) or feature axis (F-axis). Besides, for the e-commerce dataset, we also apply a variant of BERT, which we call eBERT, that was

trained with additional e-commerce item titles as supplement in-domain data.

3.3 Document-level Back-translation

While there exist many works showing the improvements of context-aware systems, some major aspects are typically not covered - one of them being back-translation (Sennrich et al., 2016a). Back-translation is an integral part when building the strongest possible systems and is currently the best way to include monolingual data in the training of a NMT system. It uses an inverse, target-to-source, MT model to generate synthetic source sentences given target-side monolingual sentences. There exists a series of works on this topic (Hoang et al., 2018; Burlot and Yvon, 2018; Graça et al., 2019). However, the underlying inverse MT model used so far is mostly on the sentence level.

In this work, we argue that back-translation could be even more crucial when training document-level NMT systems, since even for common language pairs like German-English we have very limited amounts of parallel document-level data while having an abundance of monolingual document-level data. In addition, except for using a sentence-level inverse NMT model, we also introduce a document-level inverse MT model to generate pseudo source documents given monolingual target-side documents. The intuition behind this approach is that we expect the document-level back-translation system to keep more inter-sentential discourse-phenomena in the synthetic source documents. If the back-translation system is merely on the sentence level, some discourse-phenomena, like consistent lexical choices, might not remain in the generated source documents. Losing this potentially large amount of discourse-phenomena is not beneficial for training a context-aware model.

Since there is a large amount of document-level, monolingual in-domain data in the form of the NewsCrawl corpora, we conduct back-translation experiments on the WMT task. Here, we first train a baseline model and a context-aware model on the WMT news-commentary v14 in the reverse direction (De-En). We decide to use Multi-Encoders (In. Par.) as our inverse context-aware model, as it has the best performance on the WMT task. Then we sample 4.8M sentences pairs from news-docs2018 monolingual corpus,[1] which contains 168K documents. Next, we use the inverse NMT models to translate them applying beam search with beam size 5 and concatenate the resulting bilingual synthetic data with the real documents in the news-commentary v14 dataset (En-De). Finally, we compare the performance of a sentence-level baseline (En-De) and a context-aware model, Single Encoder (2to2), on both concatenated corpora. To our knowledge, this is the first attempt to explore the document-level back-translation data systematically (see Section 4.3.5).

4 Experiments

4.1 Datasets

We experiment with various parallel document-level datasets including IWSLT TED talk English-Italian,[2] WMT news-commentary v14 English-German,[3] OpenSubtitles (Lison and Tiedemann, 2016) v2018 English-German[4] and an additional in-house e-commerce English-Chinese dataset. The test sets for the former two are the IWSLT 2017 test set and WMT newstest2018, respectively; for the latter two, we have created the dev and test sets ourselves by doing appropriate splits to the complete dataset.[5] The data statistics of bilingual corpora used for fine-tuning context-aware models are summarized in Table 1. In the IWSLT, WMT and OpenSubtitles datasets, there exists a boundary between documents. We first take them as sentence-level corpora to train the baseline and further fine-tune the context-aware system on them.

The context-aware part of the e-commerce dataset is quite small and distinct from the other tasks: it does not contain documents or talks, but rather sentence-level item descriptions from an e-commerce website. As translation context, we provide the title of the item, instead of preceding sentences. Item descriptions and titles are user-provided, so they may contain ungrammatical sentences, spelling errors, and other noise. We give the title as context on the source-side, and we have reference translations only for the descriptions. In

[1] http://data.statmt.org/news-crawl/doc/de

[2] https://sites.google.com/site/iwsltevaluation2017

[3] https://www.statmt.org/wmt18/translation-task.html

[4] http://opus.nlpl.eu/OpenSubtitles-v2018.php

[5] We randomly sample complete documents from different years for dev and test set. The precise document IDs are: dev: {1997/517700, 2002/696617, 2007/933906, 2012/2192989, 2017/6007584}, test: {1997/708495, 2002/257044, 2007/1036109, 2012/2322334, 2017/6190628}

	IWSLT			WMT			OpenSubtitles			E-commerce data		
# Sentences	233K/	1.6K/	1.2K	338K/	2.2K/	3.0K	22.5M/	3.5K/	3.8K	36K/	478/	1K
# Running words	4.7M/	31K/	22K	8.3M/	47K/	68K	188M/	30K/	30K	596K/	12K/	26K
Avg. sentence length	20/	20/	19	25/	22/	23	8/	9/	8	17/	25/	26

Table 1: Training/development/test corpora statistics.

order to get a strong baseline, we additionally use a large sentence-level e-commerce dataset consisting of 6M sentence pairs (2.7M in-domain and 3.3M out-of-domain e-commerce) to train the baseline system, and then use it as initialization for fine-tuning on the context-aware e-commerce dataset. This dataset allows us to investigate context-aware NMT in a realistic scenario, in which the majority of training data does not have additional context.

To get a better insight into the model's performance for tackling the pronoun translation, we evaluate our models on two targeted test sets: one is ControPro for OpenSubtitles, the other is a coreference-focused test set for WMT. ControPro is introduced in Müller et al. (2018), which is a contrastive test set extracted from OpenSubtitles with previous sentences as context. The source sentence has the English pronoun *it* and three corresponding German translations containing German counterparts *es*, *sie*, *er*, i.e., one of them is correct, and the other two are incorrect. The evaluation is done by counting the decisions that models rank the correct translation higher than the incorrect translations. In addition to using it in this way, we keep the source and the corresponding correct translation to form a standard test set containing 12K sentence pairs and measure the general translation quality on it.

	ControPro	Coreference
# Sentences	12K	1.1K
# Running words	129K	28K
Avg. sentence length	11	26

Table 2: Two targeted-test sets: ControPro (Müller et al., 2018) and coreference-focused test set extracted from WMT newstest 2008-2019 using NeuralCoref.

To create a targeted test set for WMT, we use an external tool called NeuralCoref[6]. We first apply this external tool to detect the coreference resolution in two consecutive sentences from newstest2008-2019, and then only keep the sentences where the coreference is resolved intersententially. This results in a targeted test set containing more inter-sentential discourse phenomena. The detailed statistics of these two targeted test sets are given in Table 2.

All language pairs are preprocessed with the Moses tokenizer[7] except for the Chinese corpus which is preprocessed with the chinese text segmentation tool "jieba"[8]. We apply byte pair encoding (Sennrich et al., 2016b) with 32k merge operations jointly for source and target languages.

4.2 Experimental setting

All models are implemented in open-source toolkit OpenNMT (Klein et al., 2017). For the sentence-level baseline system, we follow a 6-layer base Transformer model (Vaswani et al., 2017) and set the hidden size and embedding size as 512 and the dimension of the feed-forward layer as 2048. We use 8 heads for multi-head attention. For our context-aware models, we extend baseline system to include additional encoder with the same setting. In training, we use Adam optimizer (Kingma and Ba, 2014) or its variant Lazy Adam Optimizer for optimization and follow the learning rate schedule described in (Vaswani et al., 2017). The learning rate scale factor and warm-up steps are different for different datasets. In all our experiments, we share word embeddings over the source and the context. The context encoders are also initialized by the encoder of the sentence-level baseline.

For automatic evaluation, we report case-sensitive sacreBLEU score (Post, 2018) for all corpora except for e-commerce, on which the evaluation is done in Chinese character-level with case-insensitive sacreBLEU.

4.3 Analysis

4.3.1 Performance in Terms of BLEU

Table 3 shows the corpus-level BLEU-scores of all architectures on different tasks. For the baseline as well as the "source-side-only" systems we get similar results to Kim et al. (2019) on the IWSLT

[6] https://github.com/huggingface/neuralcoref

[7] http://www.statmt.org/moses

[8] https://github.com/fxsjy/jieba

System	Type	IWSLT BLEU[%]	WMT BLEU[%]	OpenSubtitles BLEU[%]	E-commerce data BLEU[%]
Baseline	N/A	31.6	28.4	37.3	33.7
Single Encoder (2to1)	s	31.7	28.3	37.5	32.8
Single Encoder (3to1)	s	31.1	28.5	36.7	N/A
Multi-Encoders (Out.)	s	31.3	28.6	37.6	34.0
Multi-Encoders (In. Seq.)	s	31.8	29.2	37.5	**34.6**
Multi-Encoders (In. Par.)	s	32.2	**30.1**	37.5	34.2
WordEmb (In. Par.)	s	31.9	29.8	37.3	34.3
Single Encoder (2to2)	s,t	32.3	28.9	**38.2**	N/A
BERT sequence emb. (e,d)	s,m	**32.8**	29.0	37.4	34.0
BERT sequence emb. (e)	s,m	32.3	29.3	36.5	34.2
BERT sequence emb. (d)	s,m	32.1	29.7	36.6	34.3
BERT single emb. (T-axis)	s,m	31.7	28.7	37.6	34.5
eBERT single emb. (T-axis)	s,m	N/A	N/A	N/A	34.5
BERT single emb. (F-axis)	s,m	31.6	28.7	36.7	32.3

Table 3: Comparison of document-level architectures on different tasks. "Type" indicates whether the context used is from source(s)- or target(t)-side or if additional monolingual(m) data is included. "e" or "d" following the name of BERT sequence emb. approach indicates whether the context representation is fused on the encoder or decoder.

and WMT tasks, with Multi-Encoders (In. Par.) being the strongest architecture. For the e-commerce data, Multi-Encoders (In. Seq.) performs slightly better. Interestingly, with these architectures we do not see improvements on the much larger OpenSubtitles corpus. This seems to confirm the suggestion of Kim et al. (2019) that these architectures work more as a regularization which is much more important for low resource tasks.

The Single Encoder (2to2) results in an improvement on all tasks excluding the e-commerce task, for which the method is not applicable due to the lack of target translation of the context (titles). The improvements on the OpenSubtitles test set are comparable to what has been reported in the literature (Tiedemann and Scherrer, 2017) while the improvements on the other tasks are a bit smaller. We notice that with this architecture, the improvements increase with decreasing average sentence length, which makes sense since it is known that the Transformer struggles with long input sequences (Rosendahl et al., 2019). This seems also to be indicated by the deteriorating performance of the Single Encoder (3to1) system, which confirms the findings of Agrawal et al. (2018).

Including context information through BERT sequence embeddings improves the performance on IWSLT, WMT and the e-commerce tasks but not on OpenSubtitles. The pre-trained BERT brings more (monolingual) data, which should again help primarily on the low resource tasks. Contrary to the before mentioned approaches, the BERT single embedding approach does not significantly increase the number of free parameters, but it only works on the e-commerce task in our experiments. This finding as well as the discrepancy between concatenating along the time or feature axis is discussed in detail in Section 4.3.2.

While these findings are consistent with previous works, we find it to be insufficient to just rely on corpus-level BLEU scores to come to a conclusion about the usefulness of these approaches. In the subsequent sections we discuss specific aspects of the translations which might be easily overlooked. Furthermore we investigate the utilization of back-translation (Sennrich et al., 2016a) for document-level systems, in an effort to compare these architectures in a more "real-life" setting where back-translation is almost always used.

4.3.2 Including BERT

When looking at the results in Table 3, we see that using the embeddings produced by BERT yields some decent improvements on all tasks except for OpenSubtitles. This might indicate that the improvements - at least in parts - come from the usage of additional data when training the BERT model rather than from an improved context representation. A drawback when using the BERT system combination is the introduction of many additional parameters and calculations. This can be drasti-

System	IWSLT		WMT		E-commerce data	
	# tokens	BLEU[%]	# tokens	BLEU[%]	# tokens	BLEU[%]
Reference	19931	-	64276	-	40149	-
Baseline	-226	31.6	+1117	28.4	-2672	33.7
BERT single emb. (T-axis)	-66	31.7	+879	28.7	-2174	34.5
Random emb. (T-axis)	+19	31.5	+1557	28.7	-2177	34.7

Table 4: Using different vectors for context representation. For the reference, the number of tokens stands for the total number of target tokens in the reference. In all consecutive lines, the number stands for the difference in the number of tokens compared to the reference.

cally reduced when using a single vector extracted from BERT as described in Section 3.2. However, the results of this approach are not significantly outperforming the baseline system on any tasks except for the e-commerce data.

Surprisingly, the eBERT does yield no further improvement over the BERT variant and the concatenation along the F-axis leads to a significant degradation in performance. These two factors lead us to believe that the context information is not the decisive factor but something else. To investigate this, we replaced the BERT-generated context vector with a random vector and compared the resulting BLEU scores which are shown in Table 4.

Depicted in this table are the BLEU score as well as the number of tokens in the respective hypothesis for the IWSLT, WMT and e-commerce tasks. For replacing the real context vector we create the random vector by sampling from the uniform distribution. Looking at the results, we see that our assumption is correct: the variant using a random vector yields the same improvements as the real context vector on the e-commerce task - even though it inhabits no relevant context information.

The reason behind this becomes clear when comparing the number of tokens produced in the hypotheses: On the e-commerce task we have a noticeable problem with under translation. We argue that by increasing the length of the input sequence we inevitably increase the length of the output, leading to a longer hypothesis and consequently to a smaller brevity penalty when calculating BLEU. This effect is not present for the other tasks at hand, since there we do not have a significant effect of under translation. We note that similar results were obtained very recently by Li et al. (2020), who also see improvements when replacing the context signal with random noise. However, we conclude that the underlying effect is a different one, since we see no improvements when concatenating along the feature axis or when evaluating on a different task. In conclusion, we argue that the improvements seen by using the BERT-embeddings for context information rather comes from additional data and other effects discussed in this section, rather than from the usage of actual context information.

4.3.3 Better Headline Translation using Context

In this section we discuss another unexpected effect of using context information in the translation, namely giving the system additional information about the nature of the input. In the WMT task, both the train and test data consist of articles composed of a headline followed by a body of text, consisting of several sentences. This means the only time the system has no context information at hand, is when translating the headline of an article. We argue that the system can in fact use this information to distinguish whether the input sequence at hand is a headline or a real sentence and act accordingly. Since a headline has a very distinguishable style compared with a complete sentence, this should lead to improvements in the translation quality. To examine this hypothesis, we separate the WMT test set into two parts: One consisting only of headlines and the other one consisting only of body of texts. We then evaluate the baseline system and our strongest document-level system (Multi-Encoders (In. Par.) for WMT) separately on both sets, The results can be seen in Table 5.

We see that the translations of both sets are improved when using the document-level setup. However, the improvement on the headlines is much larger (+4.5% BLEU) than on the body of text (+1.7% BLEU). When manually checking the hypotheses, we find that the baseline system often times tries to translate a headline as a "complete" sentence (e.g. including a verb) while the document level system translates these in a much more consistent style. This observation coincides with the fact

System	BLEU[%]
Baseline	28.4
Doc-level	**30.1**
Baseline_headlines	19.9
Doc-level_headlines	**24.4**
Baseline_newsbody	28.5
Doc-level_newsbody	**30.2**

Table 5: System performance in terms of BLEU on headlines vs body of text for the WMT test set. The document-level system is Multi-Encoders (In. Par.).

that the baseline system shows severe signs of over-translation (on average 14.9% more tokens than the reference) and the document-level system does not (-1.2%). We note that this effect is not responsible for the overall improvement in the corpus-level BLEU, since the ratio of headlines to text is very small (3.9%). This becomes clear when comparing the improvements on the body of text vs the complete test set - which is equal. We conclude that this is another instance where the context improves the translation quality even if it is not immediately obvious.

4.3.4 Pronoun Resolution

Testing the correct translation of pronouns is an established method to compare the context-awareness of document-level machine translation systems (Guillou et al., 2016; Jean et al., 2017; Bawden et al., 2018; Voita et al., 2018; Werlen et al., 2018; Wong et al., 2020). It can be argued that the ability of correctly translating inter-sentential pronouns not only depends on the architecture at hand but also on the data which the system is trained on. We decide to test the pronoun resolution capabilities of our systems in two different ways: First we are using an automatic metric for the accuracy of pronoun translation (APT) (Werlen and Popescu-Belis, 2017) and second we use two targeted test sets described in Section 4.1. The results on OpenSubtitles and WMT can be found in Table 6.

We calculate BLEU and APT scores on both the OpenSubtitles test set and ControPro test set (without contrastive translations) (Müller et al., 2018). Furthermore we calculate the resolution accuracy on ControPro (with contrastive translations). We compare the sentence-level baseline system with the best performing document-level system on this task - Single Encoder (2to2) as well as the Single Encoder (2to1) system. Even though the latter does not significantly improve over the baseline on the OpenSubtitles test set, we find a significant increase in pronoun translation accuracy in terms of both evaluation methods. The Single Encoder (2to2) system is even stronger in terms of pronoun translation, outperforming the baseline system by an impressive 33.9% absolute accuracy on the targeted test set. When calculating BLEU on ControPro, the gap between the baseline and the document-level systems becomes significantly larger. The BLEU scores for the Single Encoder (2to2) and the Single Encoder (2to1) systems are equal.

When looking at the APT scores on WMT test set, the context-aware system does not provide much improvement. We assume the reason is that the portion of the potential improvement regarding inter-sentential pronoun resolution is quite small, having looked through this test set. The increased gap of APT score between the baseline system and the context-aware system on the coreference-focused test set confirms this assumption, as there are more inter-sentential coreference phenomena in this targeted test set. Note that the BLEU score gaps between the baseline and context-aware systems on both test sets are almost the same.

All in all we can conclude that in this case the context information is helpful for a better translation, even though the effect might not be visible when just looking at corpus level BLEU.

4.3.5 Document-level Back-translation

When dealing with document-level monolingual data, the question arises, whether a sentence-level back-translation system is sufficient to generate the synthetic data. In this section, we investigate the effect of the sentence-level back-translation data and document-level back-translation data on the baseline system as well as a context-aware system. The sentence-level baseline and context-aware model used to generate synthetic documents have 28.3% BLEU and 29.7% BLEU on the test set, respectively. The performance of the resulting En-De systems are summarized in Table 7.

When using the synthetic data generated by the sentence-level system we see a huge increase in performance for both systems (+5.5% BLEU for the sentence-level system and +7.2% BLEU for the document-level system). A large increase in performance is to be expected since we increase the amount of data by roughly a factor of 8. The systems trained on the synthetic data generated by the document-level system show even further improvements (+1.6% BLEU for the sentence-level

System	OpenSubtitles		ControPro			WMT		Coreference test	
	BLEU	APT	BLEU	APT	corr.	BLEU	APT	BLEU	APT
Baseline	37.3	52.8	30.5	35.4	48.7	28.4	40.6	18.9	24.0
Single Encoder (2to1)	37.5	53.4	**33.1**	47.4	64.3	28.3	40.8	19.0	25.6
Single Encoder (2to2)	**38.2**	**54.2**	**33.1**	**49.5**	**82.6**	**28.9**	**41.1**	**19.7**	**26.1**

Table 6: Targeted evaluation for OpenSubtitles and WMT. All numbers are in percentage.

BT-Data used	System	BLEU[%]
-	Sent-level	28.4
	Doc-level	28.9
Sent-level	Sent-level	33.9
	Doc-level	36.1
Doc-level	Sent-level	35.5
	Doc-level	**36.5**

Table 7: Including back-translation data to the WMT task. The architecture of the document-level system is the Single Encoder (2to2) approach.

system and +0.4% BLEU for the document-level system). This might be in part due to the fact that the document-level back-translation system is stronger than the sentence-level one.

A very interesting observation is that the document-level system profits significantly more from the synthetic data in both scenarios. This contradicts the proposition that document-level architectures function mainly as a form of regularization for low resource data-settings. To the contrary we see an especially large gap in the case where we use only the sentence-level back-translation system for synthetic data generation. We argue that the reason for this is, that the document-level system is more capable in recovering from errors made during the back-translation due to the context information. For example a wrongly translated pronoun on the source side will definitely lead the sentence-level system astray, but the document-level one might still recover when the context is correct. This assumptions is also supported by the fact that the gap between sentence-level and document-level system gets smaller when using synthetic data generated by the document-level system, since we assume less such errors get made by this system.

5 Conclusion

In this work, we give a comprehensive comparison of current approaches to document-level NMT. To draw meaningful conclusions, we report results for standard NMT metrics on four diverse tasks - differing in the domain and the data size. We find that there is no single best approach to document-level NMT, but rather that different architectures work the best on various tasks. Looking at task-specific problems, such as pronoun resolution or headline translation, we find improvements in the context-aware systems, which is not visible in the corpus-level metric scores.

We also investigate methods to include document-level monolingual data on both source (using pre-trained embeddings) and target (using back-translation) sides. We argue that the performance improvements from the pre-trained encoder predominantly come from increased training data and other task-specific phenomena unrelated to actual context information utilization. Regarding back-translation, we find that document-level systems seem to benefit more from synthetically generated data than their sentence-level counterparts. We discuss that this is because document-level systems are more robust to sentence-level noise.

We plan to expand our experiments to incorporate document-level monolingual data on both source and target sides. This makes sense just by looking at the data conditions of almost every task: document-level parallel data is scarce, but document-level monolingual data is abundant.

Acknowledgments

Christian Herold and Yingbo Gao have received funding from the European Research Council (ERC) (under the European Union's Horizon 2020 research and innovation programme, grant agreement No 694537, project "SEQCLAS") and the Deutsche Forschungsgemeinschaft (DFG; grant agreement NE 572/8-1, project "CoreTec") and eBay Inc. The work reflects only the authors' views and none of the funding agencies is responsible for any use that may be made of the information it contains.

References

Ruchit Rajeshkumar Agrawal, Marco Turchi, and Matteo Negri. 2018. Contextual handling in neural machine translation: Look behind, ahead and on both sides. In *21st Annual Conference of the European Association for Machine Translation*, pages 11–20.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Rachel Bawden, Rico Sennrich, Alexandra Birch, and Barry Haddow. 2018. Evaluating discourse phenomena in neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1304–1313.

Franck Burlot and François Yvon. 2018. Using monolingual data in neural machine translation: a systematic study. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 144–155.

Marine Carpuat. 2009. One translation per discourse. In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions (SEW-2009)*, pages 19–27.

Junxuan Chen, Xiang Li, Jiarui Zhang, Chulun Zhou, Jianwei Cui, Bin Wang, and Jinsong Su. 2020. Modeling discourse structure for document-level neural machine translation. *arXiv preprint arXiv:2006.04721*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Miguel Graça, Yunsu Kim, Julian Schamper, Shahram Khadivi, and Hermann Ney. 2019. Generalizing back-translation in neural machine translation. In *ACL 2019 Fourth Conference on Machine Translation*, Florence, Italy. [slides].

Liane Guillou, Christian Hardmeier, Ekaterina Lapshinova-Koltunski, and Sharid Loáiciga. 2018. A pronoun test suite evaluation of the english–german MT systems at WMT 2018. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 570–577.

Liane Guillou, Christian Hardmeier, Preslav Nakov, Sara Stymne, Jörg Tiedemann, Yannick Versley, Mauro Cettolo, Bonnie Webber, and Andrei Popescu-Belis. 2016. Findings of the 2016 WMT shared task on cross-lingual pronoun prediction. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 525–542.

Christian Hardmeier. 2014. *Discourse in statistical machine translation*. Ph.D. thesis, Acta Universitatis Upsaliensis.

Christian Hardmeier and Marcello Federico. 2010. Modelling pronominal anaphora in statistical machine translation. In *IWSLT (International Workshop on Spoken Language Translation); Paris, France; December 2nd and 3rd, 2010.*, pages 283–289.

Christian Hardmeier, Preslav Nakov, Sara Stymne, Jörg Tiedemann, Yannick Versley, and Mauro Cettolo. 2015. Pronoun-focused MT and cross-lingual pronoun prediction: Findings of the 2015 DiscoMT shared task on pronoun translation. In *Proceedings of the Second Workshop on Discourse in Machine Translation*, pages 1–16.

Vu Cong Duy Hoang, Philipp Koehn, Gholamreza Haffari, and Trevor Cohn. 2018. Iterative back-translation for neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 18–24.

Sebastien Jean, Stanislas Lauly, Orhan Firat, and Kyunghyun Cho. 2017. Does neural machine translation benefit from larger context? *arXiv preprint arXiv:1704.05135*.

Marcin Junczys-Dowmunt. 2019. Microsoft translator at WMT 2019: Towards large-scale document-level neural machine translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 225–233.

Yunsu Kim, Duc Thanh Tran, and Hermann Ney. 2019. When and why is document-level context useful in neural machine translation? In *Proceedings of the Fourth Workshop on Discourse in Machine Translation (DiscoMT 2019)*, pages 24–34.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72.

Samuel Läubli, Rico Sennrich, and Martin Volk. 2018. Has machine translation achieved human parity? a case for document-level evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4791–4796.

Ronan Le Nagard and Philipp Koehn. 2010. Aiding pronoun translation with co-reference resolution. In

Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR, pages 252–261.

Bei Li, Hui Liu, Ziyang Wang, Yufan Jiang, Tong Xiao, Jingbo Zhu, Tongran Liu, and Changliang Li. 2020. Does multi-encoder help? a case study on context-aware neural machine translation. *arXiv preprint arXiv:2005.03393*.

Pierre Lison and Jörg Tiedemann. 2016. Opensubtitles2016: Extracting large parallel corpora from movie and TV subtitles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 923–929.

Sameen Maruf and Gholamreza Haffari. 2018. Document context neural machine translation with memory networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1275–1284.

Sameen Maruf, André FT Martins, and Gholamreza Haffari. 2019. Selective attention for context-aware neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3092–3102.

Ruslan Mitkov. 1999. Introduction: special issue on anaphora resolution in machine translation and multilingual NLP. *Machine translation*, pages 159–161.

Mathias Müller, Annette Rios Gonzales, Elena Voita, and Rico Sennrich. 2018. A large-scale test set for the evaluation of context-aware pronoun translation in neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 61–72.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191.

Jan Rosendahl, Viet Anh Khoa Tran, Weiyue Wang, and Hermann Ney. 2019. Analysis of positional encodings for neural machine translation. *IWSLT, Hong Kong, China*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3104–3112. Curran Associates, Inc.

Jörg Tiedemann and Yves Scherrer. 2017. Neural machine translation with extended context. In *Proceedings of the Third Workshop on Discourse in Machine Translation*, pages 82–92.

Zhaopeng Tu, Yang Liu, Shuming Shi, and Tong Zhang. 2018. Learning to remember translation history with a continuous cache. *Transactions of the Association for Computational Linguistics*, 6:407–420.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Elena Voita, Rico Sennrich, and Ivan Titov. 2019a. Context-aware monolingual repair for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 876–885.

Elena Voita, Rico Sennrich, and Ivan Titov. 2019b. When a good translation is wrong in context: Context-aware machine translation improves on deixis, ellipsis, and lexical cohesion. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1198–1212.

Elena Voita, Pavel Serdyukov, Rico Sennrich, and Ivan Titov. 2018. Context-aware neural machine translation learns anaphora resolution. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1264–1274.

Longyue Wang, Zhaopeng Tu, Andy Way, and Qun Liu. 2017. Exploiting cross-sentence context for neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2826–2831.

Lesly Miculicich Werlen and Andrei Popescu-Belis. 2017. Validation of an automatic metric for the accuracy of pronoun translation (APT). In *Proceedings of the Third Workshop on Discourse in Machine Translation*, pages 17–25.

Lesly Miculicich Werlen, Dhananjay Ram, Nikolaos Pappas, and James Henderson. 2018. Document-level neural machine translation with hierarchical attention networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2947–2954.

KayYen Wong, Sameen Maruf, and Gholamreza Haffari. 2020. Contextual neural machine translation improves translation of cataphoric pronouns. *arXiv preprint arXiv:2004.09894*.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 1480–1489.

Jiacheng Zhang, Huanbo Luan, Maosong Sun, Feifei Zhai, Jingfang Xu, Min Zhang, and Yang Liu. 2018. Improving the transformer translation model with document-level context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 533–542.

Jinhua Zhu, Yingce Xia, Lijun Wu, Di He, Tao Qin, Wengang Zhou, Houqiang Li, and Tie-Yan Liu. 2020. Incorporating BERT into neural machine translation. *arXiv preprint arXiv:2002.06823*.

A Study of Residual Adapters for Multi-Domain Neural Machine Translation

MinhQuang Pham[†‡], **Josep Crego**[†], **François Yvon**[‡], **Jean Senellart**[†]
[†]SYSTRAN / 5 rue Feydeau, 75002 Paris, France
firstname.lastname@systrangroup.com
[‡]Université Paris-Saclay, CNRS, LIMSI, 91405 Orsay, France
firstname.lastname@limsi.fr

Abstract

Domain adaptation is an old and vexing problem for machine translation systems. The most common and successful approach to supervised adaptation is to fine-tune a baseline system with in-domain parallel data. Standard fine-tuning however modifies all the network parameters, which makes this approach computationally costly and prone to overfitting. A recent, lightweight approach, instead augments a baseline model with supplementary (small) adapter layers, keeping the rest of the model unchanged. This has the additional merit to leave the baseline model intact and adaptable to multiple domains. In this paper, we conduct a thorough analysis of the adapter model in the context of a multidomain machine translation task. We contrast multiple implementations of this idea using two language pairs. Our main conclusions are that residual adapters provide a fast and cheap method for supervised multi-domain adaptation; our two variants prove as effective as the original adapter model and open perspective to also make adapted models more robust to label domain errors.

1 Introduction

Owing to multiple improvements, Neural Machine Translation (NMT) (Kalchbrenner and Blunsom, 2013; Sutskever et al., 2014; Bahdanau et al., 2015; Vaswani et al., 2017) nowadays delivers useful outputs for many language pairs. However, as many deep learning models, NMT systems need to be trained with sufficiently large amounts of data to reach their best performance. Therefore, the quality of the translation of NMT models is still limited in low-resource language or domain conditions (Duh et al., 2013; Zoph et al., 2016; Koehn and Knowles, 2017). While many approaches have been proposed to improve the quality of NMT models in low-resource domains (see the recent survey of Chu and Wang (2018)), full fine-tuning (Luong and Manning, 2015; Neubig and Hu, 2018) of a generic baseline model remains the dominant supervised approach when adapting NMT models to specific domains.

Under this view, building adapted systems is a two-step process: (a) one first trains NMT with the largest possible parallel corpora, aggregating texts from multiple, heterogeneous sources; (b) assuming that in-domain parallel documents are available for the domain of interest, one then adapts the pre-trained model by resuming training with the sole in-domain corpus. It is a conjecture that the pretrained model constitutes a better initialization than a random one, especially when adaptation data is scarce. Indeed, studies of transfer learning for NMT such as Artetxe et al. (2020); Aji et al. (2020) have confirmed this claim in extensive experiments. Full fine-tuning, that adapts all the parameters of a baseline model usually significantly improves the quality of the NMT for the chosen domain. However, it also yields large losses in translation quality for other domains, a phenomenon referred to as "catastrophic forgetting" in the neural network literature (McCloskey and Cohen, 1989). Therefore, a fully fine-tuned model is *only useful to one target domain*. As the number of domains to handle grows, training, and maintaining a separate model for each task can quickly become tedious and resource-expensive.

Several recent studies (e.g. (Vilar, 2018; Wuebker et al., 2018; Michel and Neubig, 2018; Bapna and Firat, 2019)) have proposed more lightweight schemes to perform domain adaptation, while also preserving the value of pre-trained models. Our main inspiration is the latter work, whose proposal relies on small *adapter components* that are plugged in each hidden layer. These adapters are trained only with the in-domain data, keeping the pre-trained model frozen. Because these additional

Proceedings of the 5th Conference on Machine Translation (WMT), pages 617–628
Online, November 19–20, 2020.

adapters are very small compared to the size of the baseline model, their use significantly reduces the cost of training and maintaining fine-tuned models, while delivering a performance that remains close to that of full fine-tuning.

In this paper, we would like to extend this architecture to improve NMT in several settings that still challenge automatic translation, such as translating texts from multiple topics, genre, or domains, in the face of unbalanced data distributions. Furthermore, as the notion of "domains" is not always well established, another practical setting is the translation of texts mixing several topics/domains. An additional requirement is to translate texts from domains unseen in training, based only on the unadapted system, which should then be made as strong as possible.

In this context, our main contribution is a thorough experimental study of the use of residual adapters for multi-domain translation. We notably explore ways to adjust and/or regularize adapter modules to handle situations where the adaptation data is very small. We also propose and contrast two new variants of the residual architecture: in the first one (*highway residual adapters*), adaptation still affects each layer of the architecture, but its effect is delayed till the last layer, thus making the architecture more modular and adaptive; our second variant (*gated residual adapters*) exploits this modularity and enables us to explore ways to improve performance in the face of train-test data mismatch. We experiment with two language pairs and report results that illustrate the flexibility and effectiveness of these architectures.

2 Residual adapters

In this section, we describe the basic version of the residual adapter architectures (Houlsby et al., 2019; Bapna and Firat, 2019), as well as two novel variants of this model.

2.1 Basic architecture

2.1.1 The computation of adapter layers

Our reference architecture is the Transformer model of Vaswani et al. (2017), which we assume contains a stack of layers both on the encoder and the decoder sides. Each layer contains two subparts, an attention layer, and a dense layer. Details vary from one implementation to another, we simply contend here that each layer $i \in \{1 \dots L\}$ (in the encoder or the decoder) computes a transform of a fixed-length sequence of d-dimensional input vectors h^i into a sequence of output vectors h^{i+1} as follows ($\mathbf{LN}$ denotes the (sub)layer normalization, $\mathbf{ReLU}$ is the "rectified linear unit" operator):

$$
\begin{aligned}
h_0^i &= \mathbf{LN}(h^i) \\
h_1^i &= \mathbf{W}_{db}^i h_0^i + a_1^i \\
h_2^i &= \mathbf{ReLU}(h_1^i) \\
h_3^i &= \mathbf{W}_{bd}^i h_2^i + a_2^i \\
\bar{h}^i &= h_3^i + h^i.
\end{aligned}
$$

Overall, the i^{th} adapter is thus parameterized by matrices $\mathbf{W}_{db}^i \in \mathbb{R}^{d \times b}$,$\mathbf{W}_{bd}^i \in \mathbb{R}^{b \times d}$, bias vectors $b_1^i \in \mathbb{R}^b$, $b_2^i \in \mathbb{R}^d$, with b the dimension of the adapter . For the sake of brevity, we will simply denote $h_3^i = \text{ADAP}^{(i)}(h^i)$, and $\theta_{\text{ADAP}^{(i)}}$ the corresponding set of parameters.

The "adapted" hidden vectors $\bar{h}^i_{1 \le i \le L-1}$, where L is the number of layers, will then be the input of the $(i+1)^{\text{th}}$ layer; $\bar{h}^L$ is passed to the decoder if it belongs to the encoder side, or is the input of output layer if it belongs to the decoder side. Note that zeroing out all adapters enables us to recover the basic Transformer, with $\bar{h}^i = h^i$ for all i.

In the experiments of Section 3, we use $2 \times L = 12$ residual adapters, one for each of the $L = 6$ attention layers of the encoder and similarly for the decoder.[1]

2.1.2 Design space and variants

This general architecture leaves open many design choices pertaining to the details of the network organization, the training procedure, and the corresponding objective function.

The first question is the number of adapter layers. While in principle, all Transformer layers can be subject to adaptation, it is nonetheless worthwhile to consider simpler adaptation schemes, which would only alter a limited number of layers. Such strategy might be especially relevant when the training data contains very small domains, as in the experiments of Section 3, and for which a complete adaptation may not be necessary or/and or prone to overfitting. Likewise, it might be meaningful to explore ways to share subsets of adapters across domains. This, in turn, raises the issue of which layer(s) to adapt, a question that can be approached in the light of recent analyses of Transformers models, which conjecture that the higher layers encode

[1] In the decoder, the stack of self-attention and cross encoder-decoder attention only counts as one attention layer and only corresponds to one residual adapter.

global patterns with a more "semantic" interpretation, while the lower layers encode local patterns akin to morpho-syntactic information (Raganato and Tiedemann, 2018).

A related question concerns the regularization of adapter layers to mitigate overfitting. Reducing the number of adapters, or their dimensions, is simple, but such choices are difficult to optimize numerically – an issue that becomes important as the number of domain grows. Less naive alternatives can also be entertained, such as applying weight decay or layer regularization to the adapter. Implementing these requires to modify the objective function in a way that still allows for a smooth optimization problem. For instance, weight decay applies a penalization on the weights of the adapters, complementing the cross-entropy term with a function of the norm of the parameters:

$$\bar{L} = \frac{1}{\#(x,y)} \sum_{x,y} (-\log(P(y|x))) + \lambda \sum_{i \in \{1,..,6\} \otimes \{enc,dec\}} \|\theta_{\mathrm{ADAP}^{(i)}}\|_2$$

An alternative scheme is *layer regularization*, which penalizes the output of the adapters, corresponding to the following objective:

$$\bar{L} = \frac{1}{\#(x,y)} \sum_{x,y} (-\log(P(y|x)) + \lambda \sum_{i \in \{1,..,6\} \otimes \{enc,dec\}} \|\mathrm{ADAP}^{(i)}(h_i(x,y))\|_2)$$

Finally, another independent design choice relates to the training strategy for adapters. A first option is to generalize supervised domain adaptation to multi-domain adaptation and to proceed in two steps: (a) train a generic model with all the available data; (b) train each adapter layer with domain-specific data, keeping the generic model parameters unchanged. Another strategy is to adopt the view of Dredze and Crammer (2008), where the multi-domain setting is viewed as an instance of multi-task learning (Caruana, 1997) with each domain corresponding to a specific task. This suggests training all the parameters from scratch, as we would do in a multi-task mode. The generic parameters will still depend on all the available data, while each adapter will only be trained with the corresponding in-domain data.

2.2 Highway Residual Adapters

In the basic architecture described in Section 2.1, the computation performed by lower level layers will impact all the subsequent layers. In this section, we introduce an alternative implementation of the same idea, which however delays the adaptation of each layer to the last layer (of the encoder or the decoder) as depicted on Figure 1. While the basic architecture performs adaptation in sequence, we propose here to perform it in parallel. In this version, only the last hidden vector of the encoder (decoder) is thus modified according to:

$$\bar{h}^L = h^L + \sum_{1 \le i \le L} ADAP^i(h^i) \qquad (1)$$

One obvious benefit of this variant is that it allows us to reuse the hidden vectors h^i of all hidden layers when computing an adapted output for several domains during the inference. In this situation, the forward step needs only to compute the hidden vectors h^i once for the inner encoder layers, before an adapted sequence of vectors is computed at the topmost layer. Therefore, we can fine-tune the model to multiple domains at once without recomputing h^i. This variant also opens the way to more parameter sharing across adapters, a perspective that we will not explore further in this work. Instead, we use it to develop a second variation of the adapter model, that is presented in the next section.

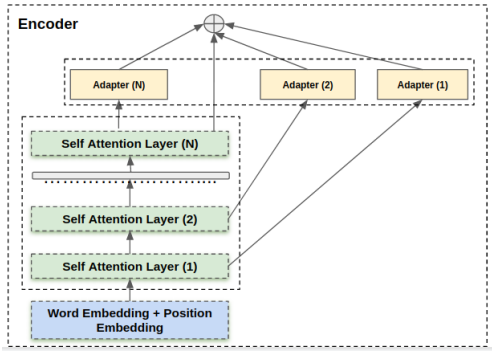


Figure 1: Highway residual adapter network

2.3 Gated Residual Adapters

The basic architecture presented above rests on a rather simplistic view of "domains" as made of well-separated and unrelated pieces of texts that are processed independently during adaptation. Likewise, when translating test documents, one needs to choose between either using one specific domain-adapted model or resorting to the generic model. In

this context, using wrong domain labels can have a strong (negative) effect on translation performance.

Therefore, we would like to design a version of residual adapters that is more robust to such domain errors. This variant, called the *gated residual adapter model*, relies on the training of a supplementary component that will help decide whether to activate, on a word per word basis, a given residual layer and to regulate the strength of this activation. To this end, we extend the highway version of residual adapters as follows.

Formally, we replace the adapter computation of equation (1) and take the adapted hidden (topmost) layer to be computed as (this is for domain k):

$$\bar{h}^L = h^L + \sum_{1 \leq i \leq L} \text{ADAP}_k^i(h^i) \odot z_k(h^L), \quad (2)$$

where the scalar $z_k(h^L[t]) \in [0, 1]$ measures the relatedness of the t^{th} word w_t to domain k. The more likely w_t is in domain k, the larger $z_k(h^L[t])$ should be; conversely, for words[2] that are not typical of any domain k (eg. function words), adaptation is minimum and the corresponding adapted encoder output ($\bar{h}^L[t]$) will remain close to the output of the generic model ($h^L[t]$). In our implementation, we incorporate two domain classifiers on top of the encoder and the decoder, that take the last hidden layer of the encoder (resp. decoder) as input and use the posterior probability $P(k|h^L[t])$ of domain k as the value for $z_k(h^L[t])$.

Training gated residual adapters thus comprises three steps, instead of two for the baseline version:

1. learn a generic model with mixed corpora from multiple domains.

2. train a domain classifier on top of the encoder and decoder; during this step, the parameters of the generic model are frozen. This model computes the posterior domain probability $P(k|h^L[t])$ for each word w_t, based on the representation computed by the last layer.

3. train the parameters of adapters with in-domain data separately for each domain, while freezing all the other parameters.

[2]The term "word" is employed here by mere convenience, as systems only manipulate sub-lexical BPE units; furthermore, the values of the hidden representations h^i at position t depend upon all the other positions in the sentence.

3 Experimental settings

3.1 Data and metrics

We perform our experiments with two translation pairs involving multiple domains: English-French (En→Fr) and English-German (En→De). For the former pair, we use texts[3] initially from 6 domains, corresponding to the following data sources: the UFAL Medical corpus V1.0 (MED)[4], the European Central Bank corpus (BANK) (Tiedemann, 2012); The JRC-Acquis Communautaire corpus (LAW) (Steinberger et al., 2006), documentations for KDE, Ubuntu, GNOME and PHP from Opus collection (Tiedemann, 2009), collectively merged in a IT-domain, Ted Talks (TALK) (Cettolo et al., 2012), and the Koran (REL). Complementary experiments also use v12 of the News Commentary corpus (NEWS). Corpus statistics are in Table 1.

En→De is a much larger task, for which we use corpora distributed for the News task of WMT20[5] including: European Central Bank corpus (BANK), European Economic and Social Committee corpus (ECO), European Medicines Agency corpus (MED)[6], Press Release Database of European Commission corpus, News Commentary v15 corpus, Common Crawl corpus (NEWS), Europarl v10 (GOV), Tilde MODEL - czechtourism (TOUR)[7], Paracrawl and Wikipedia Matrix (WEB). Statistics are in Table 2.

We randomly select in each corpus a development and a test set of 1,000 lines each and keep the rest for training.[8] Development sets help choose the best model according to the average BLEU score (Papineni et al., 2002).[9]

3.2 Baseline architectures

Using Transformers (Vaswani et al., 2017) implemented in OpenNMT-tf[10] (Klein et al., 2017), we train the following baselines:

- a generic model trained on a concatenation of all corpora, denoted **Mixed**;

[3]Most corpora are available from the Opus web site: http://opus.nlpl.eu

[4]https://ufal.mff.cuni.cz/ufal_medical_corpus

[5]http://www.statmt.org/wmt20/news.html

[6]https://tilde-model.s3-eu-west-1.amazonaws.com/Tilde_MODEL_Corpus.html

[7]https://tilde-model.s3-eu-west-1.amazonaws.com/Tilde_MODEL_Corpus.html

[8]Scripts to replicate these experiments are available at urlhttps://github.com/qmpham/experiments.git.

[9]We use truecasing and the multibleu script.

[10]https://github.com/OpenNMT/OpenNMT-tf

MED	LAW	BANK	IT	TALK	REL	NEWS
2609 (0.68)	190 (0.05)	501 (0.13)	270 (0.07)	160 (0.04)	130 (0.03)	260 (0)

Table 1: Corpora statistics for En→Fr : number of parallel lines ($\times 10^3$) and proportion in the basic domain mixture (which does not include the NEWS domain). MED is the largest domain, containing almost 70% of the sentences, while REL is the smallest, with only 3% of the data.

BANK	ECO	MED	GOV	NEWS	TOUR	WEB
4 (0.00022)	2857 (0.15)	347 (0.018)	1828 (0.095)	3696 (0.19)	7 (0.00039)	10473 (0.54)

Table 2: Corpora statistics for En→De: number of parallel lines ($\times 10^3$) and proportion in the basic domain mixture. WEB is the largest domain, containing about 54% of the sentences, while BANK and TOUR are very small.

- a fine-tuned model (Luong and Manning, 2015; Freitag and Al-Onaizan, 2016), based on the **Mixed** system, further trained on each domain with early stopping when the development BLEU score stops increasing during 3 consecutive epochs.

For all En→Fr models, we set the embeddings size and the hidden layers size to 512. Transformers use multi-head attention with 8 heads in each of the 6 layers; the inner feedforward layer contains 2,048 cells. Residual adapters additionally use an adaptation block in each layer, composed of a 2-layer perceptron, with an inner ReLU activation function operating on normalized entries of dimension $b = 1024$. Bapna and Firat (2019) showed that the performance of adapted models increases with respect to the size of the inner dimension and obtained performance close to the full fine-tuned model with $b = 1024$, which is twice as large as the dimension of a Transformer layer. We used the same setting in our experiments.

Training uses a batch size of 12,288 tokens; optimization uses Adam with parameters $\beta_1 = 0.9$, $\beta_2 = 0.98$ and Noam decay ($warmup\_steps = 4,000$), and a dropout rate of 0.1 for all layers. For the **Mixed** model, we use an initial learning rate of 1.0 and take the concatenation of the validation sets of 6 domains for development. In the fine-tuning experiments, we continue training using **Mixed** as starting point, using the same learning rate schedule, and continuing the incrementation of the number of steps. In the multi-task training, we use the same learning rate schedule as for **Mixed**: for each iteration, we sample a domain a probability proportional to its size; we then sample a batch of 12,288 tokens that is used to update the shared parameters and the parameters of the corresponding adapter.

Models for En→De are larger and rely on embeddings as well as hidden layers of size 1024; each Transformers layer contains 16 attention heads; the inner feedforward layer contains 4,096 cells. Adapter modules have the same architecture as for the other language pair, except for their size, which is doubled ($b = 2,048$).

3.3 Multi-domain systems

In this section, we evaluate several proposals from the literature on multi-domain adaptation and compare them to full fine-tuning on the one hand, and to two variants of the residual adapter architecture on the other hand. The reference methods included in our experiments are the following:

- a system using "domain control" (Kobus et al., 2017). In this approach, domain information is introduced either as an additional token for each source sentence (**DC-Tag**) or in the form of a supplementary feature for each word (**DC-Feat**);

- a system using lexicalized domain representations (Pham et al., 2019): word embeddings are composed of a generic and a domain-specific part (**LDR**);

- the three proposals of Britz et al. (2017). **TTM** is a feature-based approach where the domain tag is introduced as an extra word *on the target side*. The training uses reference tags and inference is performed with predicted tags, just like for regular target words. **DM** is a multi-task learner where a domain classifier is trained on top of the MT encoder, so as to make it aware of domain differences; **ADM** is the adversarial version of **DM**, pushing the encoder towards learning domain-independent source representations. These methods only use domain labels in training.

Model / Domain	MED	LAW	BANK	TALK	IT	REL	AVG
Mixed	37.3	54.6	50.1	33.5	43.2	77.5	49.4
FT-Full	37.7	59.2	54.5	34.0	46.8	90.8	53.8
DC-Tag	38.1	55.3	49.9	33.2	43.5	80.5	50.1
DC-Feat	37.7	54.9	49.5	32.9	43.6	79.9	49.9
LDR	37.0	54.7	49.9	33.9	43.6	79.9	49.8
TTM	37.3	54.9	49.5	32.9	43.6	79.9	49.7
DM	35.6	49.5	45.6	29.9	37.1	62.4	43.4
ADM	36.4	53.5	48.3	32.0	41.5	73.4	47.5
Res-Adap	37.3	57.9	53.9	33.8	46.7	90.2	53.3
Res-Adap-MT	37.9	56.0	51.2	33.5	44.4	88.3	51.9
Res-Adap-MT$^+$	37.5	57.1	52.4	33.7	46.2	89.5	52.7
Res-Adap-MT (gen)	37.7	51.0	34.0	30.4	34.2	15.2	36.4

Table 3: Translation performance of various multi-domain MT systems (En→Fr) compared to variants of the residual adapter models.

The two variants of the residual adapter model included in this first round of experiment have been presented in Section 2.1: **Res-Adap** is the multi-domain version of the approach of Bapna and Firat (2019) based on a two-step training procedure; while **Res-Adap-MT** is the "multi-task" version, where the parameters of the generic model and of the adapters are jointly learned from scratch. We also report results for the same system, using the the parameters of the **Mixed** model as initialization (**Res-Adap-MT**$^+$).[11]

Because of the limit of our computational resources, we restrict the experiments in this section to the En→Fr task. Results are in Table 3.

These results first show that full fine-tuning outperforms all other methods for the in-domain test sets. However, **Res-Adap** is able to reduce the gap with this approach for several domains, showing the effectiveness of residual adapters. The "multi-task" variant is slightly less effective in our experiments than the basic version, where optimization is performed in two steps. As it turns out, using residual adapters proves here better on average than the other reference multi-domain systems; it is also much better than the generic system for translating data from known domains, outperforming the **Mixed** system by more than 4 BLEU points in average. Gains are especially large for small domains such as LAW and REL.

Comparing training schemes (**Res-Adap** vs **Res-Adap-MT** vs **Res-Adap-MT**$^+$) suggests that the simultaneous learning of all parameters is detrimental to performance in our settings: we see that the 2-step procedure implemented in **Res-Adap** always yields the best scores, even when **Res-Adap-MT** is initialized with good parameter values . This may be because in this setting, the adapters have access to a stable version of the generic system. The last line (Res-Adap-MT (gen)) gives the results for a **Res-Adap-MT** trained system in which we cancel the adapter in inference - comparing this to **Mixed** shows how differently the generic parts of these two systems behave.

3.4 Varying the positions and number of residual adapters

Tables 4-5 report BLEU scores for 6 domains in each language pair: MED,LAW,BANK,TALK,IT and REL for En→Fr; GOV, ECO, TOUR, BANK, MED and NEWS for En→De. We first see that for the latter direction, the basic version **Res-Adap** also outperforms the **mixed** baseline on average, with large gains for the small domains TOUR, BANK and comparable results for the other domains.

By varying the number and position of residual adapters (see Section 2.1), we then contrast several implementations. Because the set of possible configurations is large, we only perform experiments for layers $i = 2, 4, 6$ (both for the encoder and decoder). Two settings are considered: keeping just one adapter or keeping the three. The trend is the same for the two language directions: suppressing adapters always hurts the overall performance, albeit by a small margin: having six adapters is better than three, which is better than keeping only one. With only one adapter active, we observe small,

[11] This system also includes a layer dropout policy that cancels adapter layers with probability 0.5

insignificant changes in performance when varying the adapter's depth.

3.5 Regularizing fine-tuning

The translation from English into German includes two domains (TOUR and BANK) that are extremely small and account only for a very small fraction of the training data (respectively for 0.039% and 0.022% of the total number of sentences). Fine-tuning on these domains can lead to serious overfitting. We assess two well-known regularization techniques for adapter modules, that could help mitigate this problem: weight decay and layer regularization.

For each method, the optimal hyper-parameter λ (weight decay or layer regularization coefficient, see Section 2.1.2) are chosen by grid search in a small set of values ($\{10^{-3}, 10^{-4}, 10^{-5}\}$).

Results in Tables 4 and 5 show that regularizing the adapter model can positively impact the test performance for the smallest domains (this is especially clear for weight-decay (**Res-Adap-WD**) in En→De), at the cost however of a small drop in performance for the other domains. Using layer regularization proves here to be comparatively less effective. Finding better ways to set the regularization parameters, for instance by varying λ for each domain based on the available supervision data, is left for future work.

3.6 Highway and Gated Residual Adaptaters

We now turn to the evaluation of our new architectural variants: Highway residual adapters **Res-Adap-HW** on the one hand, and Gated residual adapters **Res-Adap-Gated** on the other hand. We use the same domains and settings as before, focusing here exclusively on the language direction En→Fr.

To also evaluate the robustness with respect to out-of-domain examples, we perform two additional experiments. We first generate translations with erroneous (more precisely: randomly assigned) domain information: the corresponding results appear in Table 6 under column RND. We also compute translation for a domain unseen in training (NEWS) as follows. For each sentence of this test set, we automatically evaluate the closest domain,[12] then use the predicted domain label to compute the translation. This is an error-prone process, which also challenges the robustness of our multi-domain systems. Results are in Table 6.

A first observation is that for domains seen in training, our variants **Res-Adap-HW** and **Res-Adap-Gated** achieve BLEU scores that are on a par to those of the original version (**Res-Adap**), with insignificant variations across test sets.

The two other settings are instructive in several ways: they first clearly illustrate the brittleness of domain-adapted systems, for which large drops in performance (more than 15 BLEU points on average) are observed when the domain label is randomly chosen. Our gated variant however proves much more robust than the other adaptation strategy and performs almost on par to the generic system for that test condition. The same trend holds for the unseen NEWS domain, with **Res-Adap-Gated** being the best domain adapted system in our set, outperforming the other variants by about 2 BLEU points.

4 Related Work

Training with data from multiple, heterogeneous sources is a common scenario in natural language processing (Dredze and Crammer, 2008; Finkel and Manning, 2009). It is thus no wonder that the design of multi-domain systems has been proposed for many tasks. In this short survey, we exclusively focus on machine translation; it is likely that similar methods (parameter sharing, instance selection/weighting, adversarial training, etc) have also been proposed for other tasks.

Early approaches to multi-domain MT were proposed for statistical MT, either considering multiple data sources (eg. Banerjee et al. (2010); Clark et al. (2012); Sennrich et al. (2013); Huck et al. (2015)) or domains containing several topics (Eidelman et al., 2012; Hasler et al., 2014). Two main strategies emerge: feature-based methods, where domain labels are integrated through supplementary features; and instance-based methods, involving a measure of similarity between train and test domains.

The former approach has also been adapted to NMT: Kobus et al. (2017); Tars and Fishel (2018) use an additional domain feature in an RNN model, in the form of an extra domain-token or of additional domain-features associated with each word. Chen et al. (2016) apply domain control on the *target* side, using a topic vector to describe the

[12] As measured by the perplexity of a language model trained with only in-domain data..

Model / Domain	MED	LAW	BANK	TALK	IT	REL	AVG	PARAMS
Mixed	37.3	54.6	50.1	33.5	43.2	77.5	49.4	65M/0
Res-Adap	37.3	57.9	53.9	33.8	46.7	90.2	53.3	65M/12M
Res-Adap$_{(2,4,6)}$	37.7	57	53	33.3	45	90	52.7	65M/6M
Res-Adap$_{(6)}$	37.7	55.8	51.5	33.9	43.6	89.2	51.9	65M/2M
Res-Adap$_{(4)}$	37.9	55.6	51.7	33.7	44.4	88.7	52	65M/2M
Res-Adap$_{(2)}$	37.8	55.5	51.4	34	43.8	86.7	51.5	65M/2M
Res-Adap-WD	37.2	56.0	52.9	33.4	46.0	90.6	52.7	65M/12M
Res-Adap-LR	37.4	56.1	51.8	33.3	45.0	89.7	52.2	65M/12M

Table 4: Translation performance of various fine-tuned systems (En→Fr). We report BLEU scores for each domain, as well as averages across domains. Column PARAMS reports the number of domain-agnostic/domain-specific parameters.

Model / Domain	GOV	ECO	TOUR	BANK	MED	NEWS	AVG	PARAMS
Mixed	29.3	30.5	17.6	38.1	47.9	20.9	30.6	213M/0M
Res-Adap	29.6	30.4	19.2	49.0	47.2	20.6	33.1	213M/48M
Res-Adap$_{(2,4,6)}$	29.7	30.5	18.8	49.6	47.1	20.6	32.7	213M/24M
Res-Adap$_{(6)}$	29.5	30.4	18.1	49.1	46.9	20.4	32.4	213M/8M
Res-Adap$_{(4)}$	29.7	30.4	18.1	49.6	47.0	20.6	32.6	213M/8M
Res-Adap$_{(2)}$	29.6	30.4	18.3	49.4	46.7	20.6	32.5	213M/8M
Res-Adap-WD	29.7	30.8	20.4	50.2	47.7	20.6	33.2	213M/48M
Res-Adap-LR	29.6	30.4	19.2	49.0	47.2	20.6	33.1	213M/48M

Table 5: Translation performance of various fine-tuned systems (En→De). We report BLEU scores for each domain, as well as averages across domains. Column PARAMS reports the number of domain-agnostic/domain-specific parameters.

Model / Domain	MED	LAW	BANK	TALK	IT	REL	AVG	RND	NEWS
Mixed	37.3	54.6	50.1	33.5	43.2	77.5	49.4	49.4	23.5
FT-Full	37.7	59.2	54.5	34.0	46.8	90.8	53.8	32.5	20.2
Res-Adap	37.3	57.9	53.9	33.8	46.7	90.2	53.3	38.4	20.5
Res-Adap-HW	37.5	57.2	53.4	33.1	46.3	91.0	53.1	36.6	20.2
Res-Adap-HW-MT	37.4	56.4	52.1	33.7	44.8	89.8	52.4	27.1	20.4
Res-Adap-HW-MT^{+}	37.7	57.0	52.5	33.5	46.1	89.0	52.6	46.5	21.4
Res-Adap-Gate	38.0	57.5	53.0	33.5	46.0	90.1	53.0	49.0	22.5

Table 6: Translation performance of highway and gated variants for En→Fr. NEWS is excluded from the training data and considered as an out-of-domain test.

whole document context. Similar ideas are developed in Chu and Dabre (2018); Pham et al. (2019), where domain differences and similarities are enforced through parameter sharing schemes. Parameter-sharing also lies at the core of the work by Jiang et al. (2019), who consider a Transformer model containing both domain-specific and domain-agnostic heads.

Britz et al. (2017) study three general techniques to take domain information into account in training: they rely on either domain classification or domain normalization on the source or target side. A contribution of this study is an adversarial training scheme to normalize representations across domains and make the combination of multiple data sources more effective. Similar techniques (parameter sharing, automatic domain classification/normalization) are at play in Zeng et al. (2018): in this work, the lower layers of the MT use auxiliary classification tasks to disentangle domain-specific from domain-agnostic representations. These representations are first processed separately, then merged to compute the final translation.

Farajian et al. (2017); Li et al. (2018) are two recent representatives of the instance-based approach: for each test sentence, a small adaptation corpus is collected based on similarity measures and used to fine-tune a mix-domain model. As shown in the former work, also adapting the training regime on a per sentence basis is crucial to make these techniques really effective.

Finally, note that a distinct evolution of the residual adapter model of Bapna and Firat (2019) is presented in Sharaf et al. (2020), where meta-learning techniques are used to make fine-tuning more effective in a standard domain-adaptation setting.

5 Conclusion and outlook

In this paper, we have performed an experimental study of the residual adapter architecture in the context of multi-domain adaptation, where the goal is to build one single system that (a) performs well for domain seen in training, ideally as well as full fine-tuning; (b) is also able to robustly handle translations for new, unseen domains. We have shown that this architecture allowed us to quickly adapt a model to a specific domain, delivering BLEU performance than are much better than the generic, mixed domain baseline, and close the gap with the full-finetuning approach, at a modest computational cost. Several new variants have been introduced and evaluated for two language directions: if none that able to clearly surpass the baseline, residual adapter models, they provide directions for improving this model in practical settings: unbalanced data condition, noise in label domains, etc. In our future work, we would like to continue the development of the gated variant, which, it seems to us, provides a flexible and robust tool to address the various challenges of multi-domain machine translation.

Acknowledgements

This work was granted access to the HPC resources of [TGCC/CINES/IDRIS] under the allocation 2020- [AD011011270] made by GENCI (Grand Equipement National de Calcul Intensif)

References

Alham Fikri Aji, Nikolay Bogoychev, Kenneth Heafield, and Rico Sennrich. 2020. In neural machine translation, what does transfer learning transfer? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7701–7710, Online. Association for Computational Linguistics.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. On the cross-lingual transferability of monolingual representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4623–4637, Online. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations*, ICLR, San Diego, CA.

Pratyush Banerjee, Jinhua Du, Baoli Li, Sudip Kumar Naskar, Andy Way, and Josef van Genabith. 2010. Combining multi-domain statistical machine translation models using automatic classifiers. In *Proceedings of the 9th Conference of the Association for Machine Translation in the Americas*, AMTA 2010, Denver, CO, USA.

Ankur Bapna and Orhan Firat. 2019. Simple, scalable adaptation for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, EMNLP-IJCNLP, pages 1538–1548, Hong Kong, China. Association for Computational Linguistics.

Denny Britz, Quoc Le, and Reid Pryzant. 2017. Effective domain mixing for neural machine translation. In *Proceedings of the Second Conference on Machine Translation*, pages 118–126, Copenhagen, Denmark. Association for Computational Linguistics.

Rich Caruana. 1997. Multitask learning. *Machine Learning*, 28(1):41–75.

Mauro Cettolo, Christian Girardi, and Marcello Federico. 2012. Wit^3: Web inventory of transcribed and translated talks. In *Proceedings of the 16th Conference of the European Association for Machine Translation (EAMT)*, pages 261–268, Trento, Italy.

Wenhu Chen, Evgeny Matusov, Shahram Khadivi, and Jan-Thorsten Peter. 2016. Guided alignment training for topic-aware neural machine translation. In *Proceedings of the Twelth Biennial Conference of the Association for Machine Translation in the Americas*, AMTA 2012, Austin, Texas.

Chenhui Chu and Raj Dabre. 2018. Multilingual and multi-domain adaptation for neural machine translation. In *Proceedings of the 24st Annual Meeting of the Association for Natural Language Processing*, NLP 2018, pages 909–912, Okayama, Japan.

Chenhui Chu and Rui Wang. 2018. A survey of domain adaptation for neural machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics*, COLING 2018, pages 1304–1319, Santa Fe, New Mexico, USA.

Jonathan H. Clark, Alon Lavie, and Chris Dyer. 2012. One system, many domains: Open-domain statistical machine translation via feature augmentation. In *Proceedings of the Tenth Biennial Conference of the Association for Machine Translation in the Americas*, (AMTA 2012), San Diego, CA.

Mark Dredze and Koby Crammer. 2008. Online methods for multi-domain learning and adaptation. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, EMNLP, pages 689–697, Honolulu, Hawaii.

Kevin Duh, Graham Neubig, Katsuhito Sudoh, and Hajime Tsukada. 2013. Adaptation data selection using neural language models: Experiments in machine translation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 678–683, Sofia, Bulgaria. Association for Computational Linguistics.

Vladimir Eidelman, Jordan Boyd-Graber, and Philip Resnik. 2012. Topic models for dynamic translation model adaptation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 115–119, Jeju Island, Korea. Association for Computational Linguistics.

M. Amin Farajian, Marco Turchi, Matteo Negri, and Marcello Federico. 2017. Multi-domain neural machine translation through unsupervised adaptation. In *Proceedings of the Second Conference on Machine Translation*, pages 127–137, Copenhagen, Denmark.

Jenny Rose Finkel and Christopher D. Manning. 2009. Hierarchical Bayesian domain adaptation. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 602–610, Boulder, Colorado.

Markus Freitag and Yaser Al-Onaizan. 2016. Fast domain adaptation for neural machine translation. *CoRR*, abs/1612.06897.

Eva Hasler, Phil Blunsom, Philipp Koehn, and Barry Haddow. 2014. Dynamic topic adaptation for phrase-based MT. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 328–337, Gothenburg, Sweden. Association for Computational Linguistics.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799, Long Beach, California, USA. PMLR.

Matthias Huck, Alexandra Birch, and Barry Haddow. 2015. Mixed domain vs. multi-domain statistical machine translation. In *Procedings of the Machine Translation Summit*, MT Summit XV, pages 240–255, Miami Florida.

Haoming Jiang, Chen Liang, Chong Wang, and Tuo Zhao. 2019. Multi-domain neural machine translation with word-level adaptive layer-wise domain mixing. *CoRR*, abs/1911.02692.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1700–1709, Seattle, Washington, USA. Association for Computational Linguistics.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Vancouver, Canada. Association for Computational Linguistics.

Catherine Kobus, Josep Crego, and Jean Senellart. 2017. Domain control for neural machine translation. In *Proceedings of the International Conference Recent Advances in Natural Language Processing*, RANLP 2017, pages 372–378, Varna, Bulgaria.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39, Vancouver. Association for Computational Linguistics.

Xiaoqing Li, Jiajun Zhang, and Chengqing Zong. 2018. One sentence one model for neural machine translation. In *Proceedings of the 11th Language Resources and Evaluation Conference*, Miyazaki, Japan. European Language Resource Association.

Minh-Thang Luong and Christopher D. Manning. 2015. Stanford neural machine translation systems for spoken language domain. In *Proceedings of the International Workshop on Spoken Language Translation*, IWSLT, Da Nang, Vietnam.

Michael McCloskey and Neal J. Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. *Psychology of Learning and Motivation - Advances in Research and Theory*, 24(C):109–165.

Paul Michel and Graham Neubig. 2018. Extreme adaptation for personalized neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 312–318, Melbourne, Australia. Association for Computational Linguistics.

Graham Neubig and Junjie Hu. 2018. Rapid adaptation of neural machine translation to new languages. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 875–880, Brussels, Belgium. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318, Stroudsburg, PA, USA.

Minh Quang Pham, Josep-Maria Crego, Jean Senellart, and François Yvon. 2019. Generic and Specialized Word Embeddings for Multi-Domain Machine Translation. In *Proceedings of the 16th International Workshop on Spoken Language Translation*, IWSLT, page 9p, Hong-Kong, CN.

Alessandro Raganato and Jörg Tiedemann. 2018. An analysis of encoder representations in transformer-based machine translation. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 287–297, Brussels, Belgium. Association for Computational Linguistics.

Rico Sennrich, Holger Schwenk, and Walid Aransa. 2013. A multi-domain translation model framework for statistical machine translation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 832–840, Sofia, Bulgaria. Association for Computational Linguistics.

Amr Sharaf, Hany Hassan, and Hal Daumé III. 2020. Meta-learning for few-shot NMT adaptation. In *Proceedings of the Fourth Workshop on Neural Generation and Translation*, pages 43–53, Online. Association for Computational Linguistics.

Ralf Steinberger, Bruno Pouliquen, Anna Widiger, Camelia Ignat, Toma Erjavec, Dan Tufis, and Dniel Varga. 2006. The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation*, LREC'06, Genoa, Italy. European Language Resources Association (ELRA).

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pages 3104–3112, Montreal, Canada. MIT Press.

Sander Tars and Mark Fishel. 2018. Multi-domain neural machine translation. In *Proceedings of the 21st Annual Conference of the European Association for Machine Translation*, EAMT, pages 259–269, Alicante, Spain. EAMT.

Jörg Tiedemann. 2009. News from OPUS - A collection of multilingual parallel corpora with tools and interfaces. In N. Nicolov, K. Bontcheva, G. Angelova, and R. Mitkov, editors, *Recent Advances in Natural Language Processing*, volume V, pages 237–248. John Benjamins, Amsterdam/Philadelphia, Borovets, Bulgaria.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eight International Conference on Language Resources and Evaluation*, LREC'12, Istanbul, Turkey. European Language Resources Association (ELRA).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

David Vilar. 2018. Learning hidden unit contribution for adapting neural machine translation models. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 500–505, New Orleans, Louisiana.

Joern Wuebker, Patrick Simianer, and John DeNero. 2018. Compact personalized models for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language*

Processing, pages 881–886, Brussels, Belgium. Association for Computational Linguistics.

Jiali Zeng, Jinsong Su, Huating Wen, Yang Liu, Jun Xie, Yongjing Yin, and Jianqiang Zhao. 2018. Multi-domain neural machine translation with word-level domain context discrimination. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 447–457, Brussels, Belgium. Association for Computational Linguistics.

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer learning for low-resource neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1568–1575, Austin, Texas. Association for Computational Linguistics.

Mitigating Gender Bias in Machine Translation with Target Gender Annotations

Artūrs Stafanovičs*†‡ and **Toms Bergmanis***†‡ and **Mārcis Pinnis**†‡

†Tilde / Vienības gatve 75A, Riga, Latvia
‡Faculty of Computing, University of Latvia / Raiņa bulv. 19, Riga, Latvia
{firstname.lastname}@tilde.lv

Abstract

When translating "*The secretary asked for details.*" to a language with grammatical gender, it might be necessary to determine the gender of the subject "*secretary*". If the sentence does not contain the necessary information, it is not always possible to disambiguate. In such cases, machine translation systems select the most common translation option, which often corresponds to the stereotypical translations, thus potentially exacerbating prejudice and marginalisation of certain groups and people. We argue that the information necessary for an adequate translation can not always be deduced from the sentence being translated or even might depend on external knowledge. Therefore, in this work, we propose to decouple the task of acquiring the necessary information from the task of learning to translate correctly when such information is available. To that end, we present a method for training machine translation systems to use word-level annotations containing information about subject's gender. To prepare training data, we annotate regular source language words with grammatical gender information of the corresponding target language words. Using such data to train machine translation systems reduces their reliance on gender stereotypes when information about the subject's gender is available. Our experiments on five language pairs show that this allows improving accuracy on the WinoMT test set by up to 25.8 percentage points.

1 Introduction

Most modern natural language processing (NLP) systems learn from natural language data. Findings of social sciences and corpus linguistics, however, indicate various forms of bias in the way humans use language (Coates, 1987; Butler, 1990; Fuertes-Olivera, 2007; Rickford, 2016). Thus the resulting NLP resources and systems also suffer from the same socially constructed biases, as well as inaccuracies and incompleteness (Jørgensen et al., 2015; Hovy and Søgaard, 2015; Prates et al., 2019; Vanmassenhove et al., 2019; Bordia and Bowman, 2019; Davidson et al., 2019; Tan and Celis, 2019). Due to the prevalent use of NLP systems, their susceptibility to social biases becomes an increasingly significant concern as NLP systems not only reflect the biases learned but also amplify and perpetuate them further (Hovy and Spruit, 2016; Crawford, 2017; HLEG, 2019).

This work concerns mitigating the manifestations of gender bias in the outputs of neural machine translation (NMT) systems in scenarios where the source language does not encode the information about gender that is required in the target language. An example is the translation of the English sentence "*The secretary asked for details.*" into Latvian. In English, the gender of "*secretary*" is ambiguous. In Latvian, however, there is a choice between the masculine noun "*sekretārs*" and the feminine noun "*sekretāre*". In cases when sentences do not contain the necessary information, NMT systems opt for translations which they have seen in training data most frequently. Acquiring the necessary information, however, might require analysis of the text beyond the level of individual sentences or require incorporation of external knowledge.

Falling back to biases, however, happens not only in the absence of the required information as NMT systems produce stereotyped translations even when clues about the subject's correct gender are present in the sentence (Stanovsky et al., 2019). This is in line with findings by Vanmassenhove et al. (2019) who suggest that NMT systems produce biased outputs not only because of the biases

*First authors with equal contribution.

Proceedings of the 5th Conference on Machine Translation (WMT), pages 629–638
Online, November 19–20, 2020.

present in data but also due to their tendency to exacerbate them.

To provide means for incorporation of external and explicit gender information, we propose a method for training NMT systems to use word-level gender annotations. To prepare training data, we project grammatical gender information of regular target language words onto the corresponding source language words. Albeit in some cases redundant, we expect that the grammatical gender information contains a useful learning signal that helps narrowing down the lexical choice of the correct target translation. As a result, the NMT system learns to rely on these annotations when and where they are available. In particular, in experiments on five language pairs, we show that the methods proposed here can be used in tandem with off-the-shelf co-reference resolution tools to improve accuracy on the WinoMT challenge set (Stanovsky et al., 2019) by up to 25.8 percentage points.

1.1 Related work

Recent recommendations for ethics guidelines for trustworthy AI recommend removing socially constructed biases at the source, the training data, prior to model training (HLEG, 2019). An example of work on debiasing training data is Zhao et al. (2018) where authors identified sentences containing animate nouns and changed their grammatical gender to the opposite. Zmigrod et al. (2019) take it further by ensuring that not only the animate nouns but also the rest of the sentence is reinflected from masculine to feminine (or vice-versa), thus preserving the morpho-syntactic agreement of the whole sentence. The applicability of this line of work is still to be established as reinflecting sentences with co-references or pairs of parallel sentences in NMT pose an additional challenge.

A different take on addressing gender biases in NMT outputs is the work on alternative generation: given a gender-ambiguous source sentence and its translation, provide an alternative translation using the opposite gender. Habash et al. (2019) approach this as a gender classification and reinflection task for target language sentences to address the first person singular cases when translating from English into Arabic. Bau et al. (2018) analyze trained NMT models to identify neurons that control various features, including gender information, that are used to generate the target sentence. In practice, however, such solutions are limited to simple source sentences where only one alternative in the target language is possible.

A complementary approach is addressing gender bias in NMT as a problem of domain mismatch. When translating TED talks, Michel and Neubig (2018) propose to adapt the NMT model for each speaker's attributes, thus also implicitly addressing previously poorly translated first-person singular cases. Saunders and Byrne (2020) describe methods for NMT model adaptation using a hand-crafted gender-balanced dataset and a translation re-scoring scheme based on the adapted models.

The closest line of work to ours is the work on the incorporation of external gender information in the NMT input. Elaraby et al. (2018) and Vanmassenhove et al. (2018) prepend training data sentences with speaker gender information to improve spoken language translation when translating into languages with grammatical gender. Moryossef et al. (2019) undertakes a similar approach at the inference time using phrases (e.g. *"she said:"*) that imply the speaker's gender. The methods proposed in this work differ from the previous work in terms of annotation granularity: we propose to use token level annotations, while the previous work used one annotation per sentence. As our training data annotations are solely based on grammatical gender, preparing them does not require any external gender information. Thus our approach is also simpler in terms of training data preparation compared to the previous work (Elaraby et al., 2018; Vanmassenhove et al., 2018).

Social Impact We propose methods to mitigate the manifestations of gender bias in the outputs of NMT. Specifically, these methods provide explicit means to incorporate information about subjects referential or social gender in NMT, thus reducing gender-based stereotyping when translating into languages which encode for grammatical gender in animate nouns. An example of a use case and a beneficiary group is the translation of occupational nouns into languages which mark gender and people for whom stereotypes of their profession do not align with their gender. While these methods can relieve gender-based representational harms by reducing stereotyped translations, they, unfortunately, provide no means for better representation of non-binary gender identities.

2 Methods

When translating from languages without grammatical gender to languages with grammatical gender,

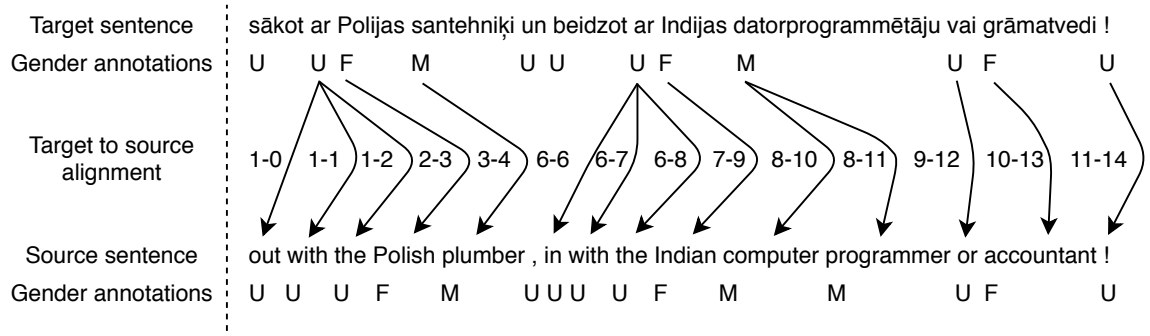


Figure 1: Illustration of target to source projections of grammatical gender annotations. Sample sentences taken from the English-Latvian development set of the WMT2017 News Translation Task.

certain words in the source sentence may not contain all the necessary information to produce an adequate and accurate translation. Examples are pronouns (e.g. *I, me, they, them, themselves*), animate nouns such as job titles and proper nouns such as names and surnames, which depending on the sentence context can be ambiguous and consequently can be translated poorly. Previous work has also shown that NMT systems are better at translating sentences that align with socially constructed gender stereotypes because they are more frequently seen in training data (Stanovsky et al., 2019; Prates et al., 2019).

To circumvent the degradation of NMT outputs due to 1) socially constructed biases and 2) absence of necessary information, we propose a method for training NMT systems to be aware of and use word-level target gender annotations (**TGA**). For training, we use data where regular source language words are annotated with the grammatical gender of their target language translations. We obtain such data by, first, morphologically tagging target language sentences to obtain information about their grammatical gender—`F` for feminine, `M` for masculine, `N` for neuter, and `U` for cases where grammatical gender is unavailable. Then, we use word-level statistical alignments to project this information from the target language to the source language words (see Figure 1 for an illustration). We use source-side factors (Sennrich and Haddow, 2016) to integrate the projected annotations as an additional input stream of the NMT system. To ensure that the NMT systems are capable of producing adequate translations when gender annotations are not available—a frequently expected case at the test time—we apply TGA dropout. We do so by randomly replacing annotations for a random number of words with U.

While useful for animate nouns, such annotations might seem otherwise redundant because the majority of nouns in training data can be expected to be inanimate. However, for some inanimate nouns, the target language grammatical gender annotations can help narrowing down the lexical choice during training. An example is the translation of *"injury"* into Latvian, where "*injury*|`F`" would result in "*trauma*" while "*injury*|`M`" would correspond to "*ievainojums*". Besides disambiguating animate nouns, annotations also disambiguate the grammatical gender of pronouns, proper nouns. Furthermore, grammatical gender annotations also concern adjectives and verbs, which in some languages have to agree in gender with the nouns they describe. Consequently, we expect that during training the NMT model will learn to use these annotations, as they contain valuable information about words in the target sentence.

At inference time, we lean heavily on the observation that there the grammatical gender of animate nouns, pronouns, and proper nouns, and the intended referential gender coincide considerably. This is, however, a heuristic and not a rule (see Hellinger and Motschenbacher (2015) for counterexamples). Nevertheless, we assume that it is possible to use TGA in a referential sense of gender, thus injecting the NMT model with additional information about the subject's gender. Sources of such information can vary; in this paper, we showcase how to use TGA together with off-the-shelf co-reference resolution tools.

2.1 Evaluation: WinoMT Test Suite

To measure the extent to which gender annotations reduce NMT systems' reliance on gender stereotypes, we use the WinoMT test suite (Stanovsky et al., 2019). WinoMT builds on the previous work

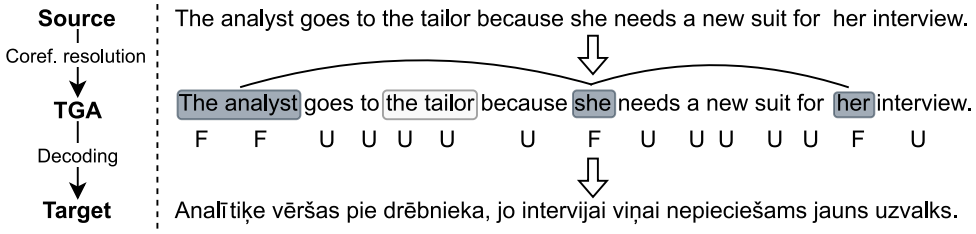


Figure 2: WinoMT test suite translation process with TGA distilled from the output of automatic coreference resolution tool.

on addressing gender bias in co-reference resolution by combining Winogender (Rudinger et al., 2018) and WinoBias (Zhao et al., 2018) datasets in a test suite for automatic evaluation of gender bias in MT. All sentences in the WinoMT test set follow the Winograd Schema where anaphora resolution is required to find an antecedent for an ambiguous pronoun (Hirst, 1981). In the case of datasets designed for evaluation of gender bias, the ambiguous pronoun refers to one of two entities which are referred to using titles of their professions. Professions and pronouns are chosen so that they either align with or diverge from the gender stereotypes of each profession as reported by the U.S. Bureau of Labor Statistics (Zhao et al., 2018).

WinoMT tests if the grammatical gender of the translation of an antecedent matches the gender of the pronoun in the original sentence. Testing is done by morphologically analysing the target translation and aligning it with the source sentence. The WinoMT test suite scores MT outputs using multiple metrics: **Accuracy** – the percentage of correctly translated antecedents, $\boldsymbol{\Delta G}$ – difference in F_1 score between sentences with masculine and feminine antecedents, $\boldsymbol{\Delta S}$ – difference in accuracy between the set of sentences that either align with or diverge from the gender stereotypes of each profession. Saunders and Byrne (2020) also propose to report **M:F** – ratio of translations using masculine and feminine antecedents.

3 Experimental Setting

Languages and Data In all our experiments, we choose one source language without grammatical gender and five Indo-European languages in which nouns have grammatical gender (see Table 1). For all language pairs, we use training data from WMT news translation tasks. We do the necessary cleaning and filtering with Moses (Koehn et al., 2007) pre-processing tools. To see how TGA is affected by data size, we also use much larger EN-LV proprietary data that we obtain from Tilde Data Libarary by combining all EN-LV parallel corpora. The proprietary data are pre-processed using the Tilde MT platform (Pinnis et al., 2018). Table 1 summarizes training data source and size statistics prior to adding TGA. For all systems and language pairs, we use byte pair encoding (BPE) (Gage, 1994; Sennrich et al., 2016) to prepare joint source and target language BPE sub-word vocabularies. We use 30K BPE merge operations and use a vocabulary threshold of 50.

	Source	# Sent.	News Test
EN-DE	WMT19	64.1M	2018
EN-FR	WMT15	39.1M	2015
EN-LV	Tilde	22.7M	2017
EN-LV	WMT17	4.5M	2017
EN-LT	WMT19	3.6M	2019
EN-RU	WMT17	25.0M	2015

Table 1: Training data set source and size in millions of sentences prior to adding TGA.

NMT Systems We use the default configuration of the Transformer (Vaswani et al., 2017) NMT model implementation of the Sockeye NMT toolkit (Hieber et al., 2020). The exception is the use of source-side factors (Sennrich and Haddow, 2016) with the dimensionality of 8 for systems using TGA, which changes the model's combined source embedding dimensionality from 512 to 520. We train all models using early stopping with patience of 10 based on their development set perplexity (Prechelt, 1998).

Morphological Taggers The preparation of training data with TGA and WinoMT evaluation relies on the outputs of a morphological tagger. If the tagger produces biased outputs, the TGA annotations might become too noisy to be useful. Furthermore, a biased morphological tagger

Tagger	F1 masc.	F1 fem.
Paikens et al. (2013)	98.6	98.7
Stanza	94.7	95.1
UDPipe	92.5	92.4

Table 2: Performance of morphological taggers on gender feature classification evaluated on the Universal Dependencies test set.

could also render WinoMT evaluation unreliable. Thus we first benchmark several morphological taggers on grammatical gender feature classification. We use Latvian as a development language because of the availability of lexicon-based and data-driven morphological analysis tools. Specifically, we use the Universal Dependencies [1] test set to compare two data-driven tools – the Stanza toolkit (Qi et al., 2020) and UDPipe (Straka and Straková, 2017). Additionally, we evaluate a dictionary-based morphological analyser and statistical tagger[2] by Paikens et al. (2013). Table 2 gives F-1 scores on masculine and feminine feature tagging. Results indicate that none of the taggers exhibits salient bias in their tagging performance. As the only non-neural system yields better F-1 scores than the other two systems, we further compare Stanza and the tagger by Paikens et al. (2013) in their impact on BLEU and WinoMT metrics. Results indicated that the choice of the tagger does not have a notable effect on BLEU scores. In terms of WinoMT accuracy scores, the NMT system that was trained using TGA prepared with Stanza yields an accuracy that is about 3% better than the system using the tagger by Paikens et al. (2013). Thus, in all remaining experiments, we use the Stanza tagger as it provides pre-trained models for a wide range of languages.

TGA in Training Data Preparing training data with TGA requires statistical word alignments between words of source and target language sentences and a target language morphological tagger. To obtain word alignments, we use *fast_align* (Dyer et al., 2013). To obtain grammatical gender information of target language words, we use the Stanza morphological tagger. When training NMT systems with TGA, we combine two copies of the original training data: one where all source-side factors are set to `U` and the other containing TGA.

TGA During Inference In training data, TGA annotate regular source language words with the grammatical gender information of corresponding target language words. We do not have access to the target language sentence during inference. Thus, we use co-reference resolution tools and extract the referential gender information from the source sentence instead. To do so, we first use co-reference resolution tools to obtain the co-reference graph. We then identify sub-graphs which contain gendered pronouns. Finally, we propagate the gender information within the graph and annotate the antecedents (see Figure 2). We set the annotations for the remaining unannotated words to `U`.

We use neural co-reference resolution tools by AllenNLP [3] (Lee et al., 2017) and Hugging Face[4] (based on work by Clark and Manning (2016)). We refer to these systems as **TGA AllenNLP** and **TGA HuggingFace** respectively. We also report the performance of NMT with TGA, when TGA use oracle information directly taken from WinoMT datasets and refer to these as **TGA Oracle**.

Evaluation We evaluate general translation quality using the BLEU (Papineni et al., 2002) metric evaluated over WMT test sets. To calculate BLEU, we use SacreBLEU[5] (Post, 2018) on cased, detokenized data. Reference test sets are only pre-processed using Moses punctuation normalization script[6]. We use the WinoMT test suite (Stanovsky et al., 2019) to measure gender bias of our NMT systems.

4 Results and Discussion

Results from experiments evaluating gender bias using the WinoMT test suite are provided in Table 3. First, we observe that all baseline systems show a strong bias towards generating translations using masculine forms. The EN-RU baseline system is the most biased as it produces only one translation hypothesis with a feminine antecedent for every 8.4 hypotheses containing masculine antecedents. Meanwhile the EN-DE baseline system

[1] https://github.com/UniversalDependencies/UD_Latvian-LVTB

[2] https://github.com/PeterisP/LVTagger

[3] https://github.com/allenai/allennlp

[4] https://github.com/huggingface/neuralcoref

[5] SacreBLEU hash: BLEU+case.mixed+numrefs.1+smooth.exp+tok.13a+version.1.3.6

[6] https://github.com/moses-smt/mosesdecoder/blob/master/scripts/tokenizer/normalize-punctuation.perl

	WMT Data Systems	Acc.	Δ G	Δ S	M:F
EN-DE	**Baseline**	66.7	10.2	14.4	2.6
	TGA Oracle	89.0	-4.7	1.7	1
	TGA HuggingFace	77.6	-0.1	11.9	1.6
	TGA AllenNLP	81.5	-2.0	11.1	1.4
EN-FR	**Baseline**	48.6	29.8	11.8	5.5
	TGA Oracle	81.5	1.4	2.8	1.2
	TGA HuggingFace	67.8	4.9	12.4	2
	TGA AllenNLP	74.4	1.6	10.1	1.6
EN-LV	**Baseline**	27.9	26.0	9.6	3.9
	TGA Oracle	42.7	15.9	10.3	2.9
	TGA HuggingFace	38.6	19.7	18.1	3.0
	TGA AllenNLP	39.3	18.1	18.6	2.8
EN-LT	**Baseline**	38.0	32.6	6.5	5.9
	TGA Oracle	52.8	15.2	4.0	2.7
	TGA HuggingFace	43.4	22.5	7.6	3.9
	TGA AllenNLP	47.2	17.7	5.1	3.1
EN-RU	**Baseline**	32.3	37.7	14.1	8.4
	TGA Oracle	55.9	10.6	14.0	2.5
	TGA HuggingFace	45.4	24.8	13.7	4.4
	TGA AllenNLP	51.4	17.0	15.2	3.2
	Proprietary Large Data System	**Acc.**	**Δ G**	**Δ S**	**M:F**
EN-LV	**Baseline**	42.0	27.9	16.6	4.9
	TGA Oracle	55.1	4.8	18.2	1.7
	TGA HuggingFace	46.2	13.5	24.1	2.6
	TGA AllenNLP	49.9	10.8	23.1	2.3

Table 3: Results on WinoMT test suite.

is the least biased with the M:F ratio being much lower – 2.6 (see the last column of Table 3). Our baseline systems for EN-DE, EN-FR and EN-RU language pairs, however, show comparable ΔG and WinoMT accuracy results to those reported by Stanovsky et al. (2019) for several publicly available commercial systems. These results confirm that our baselines, although being strongly biased, are not unordinary.

Results from experiments using TGA with oracle gender information show an improvement in WinoMT accuracy and ΔG for all language pairs (see Table 3 TGA Oracle). These results demonstrate that when training MT systems to use TGA reduces their reliance on gender stereotypes when information about the subject's gender is available, proving the usefulness of methods proposed here. Despite the availability of oracle gender information, none of the systems is entirely bias-free or obtains 100% accuracy. Thus methods proposed here could be combined with others, such as those proposed by Saunders and Byrne (2020), to achieve further improvements.

Effect on BLEU As expected, using TGA with reference sentence grammatical gender annotations has a positive effect on BLEU, thus confirming our hypothesis why and how the NMT system learns to rely on TGA as an additional source of information during training (see Table 4). It is equally important, however, that, when training NMT systems to use TGA, it does not degrade their performance when gender information is not necessary or is unavailable. Thus we test our systems for such cases by setting all TGA values to `U` and compare them to the baseline systems (see Table 4). To test for statistically significant differences between the results of NMT systems we use pairwise bootstrap resampling (Koehn, 2004) and significance threshold of 0.05. Results indicate no statistically significant differences between systems using uninformative TGA values and their baseline counterparts with an exception of results for EN-RU systems ($\Delta 0.4$ BLEU), which we find to be statistically significant.

Effect of Data Size To analyze gender bias and TGA performance depending on the quality and size of the training data, we use much larger EN-LV proprietary data (see Table 1) to train production-grade NMT systems and contrast them with EN-LV WMT data systems (see the two EN-LV sections in Table 3 and Table 5). First of all, we notice that although the large data baseline has higher WinoMT accuracy than the WMT data system, it has a similar ΔG. Decomposing ΔG as male and female grammatical gender F-1 scores (Table 5), however, clarifies that, although similarly skewed, the large data baseline has higher F-1 scores than the WMT data baseline. Next, we note, that larger training data size has a positive effect on the system's ability to use TGA more effectively as the large data system using TGA has a greater improvement on the two metrics measuring bias – ΔG and M:F[7] than its WMT data counterpart relative to its baseline. These findings suggest that TGA is a method that is applicable not only in small data settings but also in large data settings, such as commercial systems, for which it is even more effective.

Plugging-in Co-reference Resolution Tools Finally, we experiment with TGA using gender information provided by two off-the-shelf co-reference resolution tools, AllenNLP and Hugging Face. Re-

[7] ΔS results are not reliable or comparable when M:F ratios are large or differ by a large value. See result section of Saunders and Byrne (2020) for more discussion.

	Basline	TGA	All TGA= U
EN-DE	45.4	49.5	45.3
EN-FR	36.6	40.9	36.4
EN-LV	16.6	18.9	17.0
EN-LT	14.8	16.6	14.7
EN-RU	27.1	31.6	26.7

Table 4: Comparison of test set performance measured in BLEU for Baseline systems and systems trained using TGA. TGA: performance when using reference sentence *grammatical gender* annotations. All TGA=U: performance when all annotations set to be unknown.

	Male			Female		
	F-1	**P**	**R**	**F-1**	**P**	**R**
	WMT Data System					
Baseline	47.2	48.3	46.2	21.2	53.9	13.2
TGA Oracle	58.5	56.0	61.3	42.5	74.7	29.7
	Proprietary Large Data System					
Baseline	58.8	50.8	69.7	30.9	70.3	19.8
TGA Oracle	66.9	65.8	68.0	62.1	83.3	49.5

Table 5: Results of antecedent translation. Reporting grammatical gender F-1 score, precision (P) and recall (R) for EN-LV systems trained on WMT and proprietary large data.

sults show that using TGA with either of the tools outperforms baseline systems for all languages pairs. Furthermore, TGA with gender information provided by AllenNLP shows only a 4.5 to 7.1% drop in WinoMT accuracy compared to results when using TGA with oracle information. To put this in perspective, Saunders and Byrne (2020) required a handcrafted gender-balanced profession set and additional rescoring models, for their EN-DE system to obtain comparable WinoMT accuracy and ΔG without loss of translation quality. In contrast, the methods proposed here require tools that are readily available, making them easily applicable in practice.

5 Conclusions

We proposed a method for training MT systems to use word-level annotations containing information about the subject's gender. To prepare training data, the method requires a morphological tagger to annotate regular source language words with grammatical gender information of the corresponding target language words. During inference, annotations can be used to provide information about subjects' referential or social gender obtained by analyzing text beyond sentence boundaries or externally. In experiments with five language pairs, we showed that using such gender annotations reduces NMT systems' reliance on gender stereotypes in principle. We then further showed one way for how these findings can be used in practice by using off-the-shelf co-reference resolution tools.

The method proposed here decouples the task of acquiring the necessary gender information from the task of learning to translate correctly when such information is available. Thus system's ability to use such information can be achieved independently from its availability at training time. This allows for application-specific sources of gender information. Examples are the translation of chat or social media content, where users may choose to indicate their gender or translation of whole documents, where gender information may be obtained using annotations and anaphora resolution. Thus, we believe that the methods proposed here, will provide means to limit the propagation of gender stereotypes by NMT systems when translating into languages with grammatical gender.

The source code to reproduce our results for the publicly available data sets is published on GitHub[8].

Acknowledgements

This research was partly done within the scope of the undergraduate thesis project of the first author at the University of Latvia and supervised at Tilde.

This research has been supported by the European Regional Development Fund within the joint project of SIA TILDE and University of Latvia "Multilingual Artificial Intelligence Based Human Computer Interaction" No. 1.1.1.1/18/A/148.

References

Anthony Bau, Yonatan Belinkov, Hassan Sajjad, Nadir Durrani, Fahim Dalvi, and James R. Glass. 2018. Identifying and controlling important neurons in neural machine translation. *CoRR*, abs/1811.01157.

Shikha Bordia and Samuel R. Bowman. 2019. Identifying and reducing gender bias in word-level language models. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 7–15, Minneapolis, Minnesota. Association for Computational Linguistics.

[8] https://github.com/artursstaf/mitigating-gender-bias-wmt-2020

Judith Butler. 1990. Feminism and the subversion of identity. *New York and London: Routledge*, 3:1–25.

Kevin Clark and Christopher D Manning. 2016. Deep reinforcement learning for mention-ranking coreference models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2256–2262.

Jennifer Coates. 1987. *Women, men and language: A sociolinguistic account of gender differences in language*. Longman.

Kate Crawford. 2017. The trouble with bias. In *Conference on Neural Information Processing Systems, invited speaker*.

Thomas Davidson, Debasmita Bhattacharya, and Ingmar Weber. 2019. Racial bias in hate speech and abusive language detection datasets. In *Proceedings of the Third Workshop on Abusive Language Online*, pages 25–35.

Chris Dyer, Victor Chahuneau, and Noah A Smith. 2013. A simple, fast, and effective reparameterization of ibm model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648.

Mostafa Elaraby, Ahmed Y. Tawfik, Mahmoud Khaled, Hany Hassan, and Aly Osama. 2018. Gender aware spoken language translation applied to english-arabic. *2018 2nd International Conference on Natural Language and Speech Processing (ICNLSP)*, pages 1–6.

Pedro A Fuertes-Olivera. 2007. A corpus-based view of lexical gender in written business english. *English for Specific Purposes*, 26(2):219–234.

Philip Gage. 1994. A new algorithm for data compression. *C Users J.*, 12(2):23–38.

Nizar Habash, Houda Bouamor, and Christine Chung. 2019. Automatic gender identification and reinflection in Arabic. In *Proceedings of the First Workshop on Gender Bias in Natural Language Processing*, pages 155–165, Florence, Italy. Association for Computational Linguistics.

Marlis Hellinger and Heiko Motschenbacher. 2015. *Gender across languages*, volume 4. John Benjamins Publishing Company.

Felix Hieber, Tobias Domhan, Michael Denkowski, and David Vilar. 2020. Sockeye 2: A toolkit for neural machine translation. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 457–458, Lisboa, Portugal. European Association for Machine Translation.

Graeme Hirst. 1981. Anaphora in natural language understanding: A survey.

AI HLEG. 2019. Ethics guidelines for trustworthy ai. *High-Level Expert Group on Artificial Intelligence, Brussels*.

Dirk Hovy and Anders Søgaard. 2015. Tagging performance correlates with author age. In *Proceedings of the 53rd annual meeting of the Association for Computational Linguistics and the 7th international joint conference on natural language processing (volume 2: Short papers)*, pages 483–488.

Dirk Hovy and Shannon L. Spruit. 2016. The social impact of natural language processing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 591–598, Berlin, Germany. Association for Computational Linguistics.

Anna Jørgensen, Dirk Hovy, and Anders Søgaard. 2015. Challenges of studying and processing dialects in social media. In *Proceedings of the workshop on noisy user-generated text*, pages 9–18.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 388–395.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 188–197.

Paul Michel and Graham Neubig. 2018. Extreme adaptation for personalized neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 312–318, Melbourne, Australia. Association for Computational Linguistics.

Amit Moryossef, Roee Aharoni, and Yoav Goldberg. 2019. Filling gender & number gaps in neural machine translation with black-box context injection. In *Proceedings of the First Workshop on Gender Bias in Natural Language Processing*, pages 49–54, Florence, Italy. Association for Computational Linguistics.

Peteris Paikens, Laura Rituma, and Lauma Pretkalniņa. 2013. Morphological analysis with limited resources: Latvian example. In *Proceedings of the 19th Nordic Conference of Computational Linguistics (NODALIDA 2013)*, pages 267–277.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Mārcis Pinnis, Andrejs Vasiļjevs, Rihards Kalniņš, Roberts Rozis, Raivis Skadiņš, and Valters Šics. 2018. Tilde MT platform for developing client specific MT solutions. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Marcelo OR Prates, Pedro H Avelar, and Luís C Lamb. 2019. Assessing gender bias in machine translation: a case study with google translate. *Neural Computing and Applications*, pages 1–19.

Lutz Prechelt. 1998. Early stopping-but when? In *Neural Networks: Tricks of the trade*, pages 55–69. Springer.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108, Online. Association for Computational Linguistics.

John R Rickford. 2016. *Raciolinguistics: How language shapes our ideas about race*. Oxford University Press.

Rachel Rudinger, Jason Naradowsky, Brian Leonard, and Benjamin Van Durme. 2018. Gender bias in coreference resolution. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 8–14.

Danielle Saunders and Bill Byrne. 2020. Reducing gender bias in neural machine translation as a domain adaptation problem. Association for Computational Linguistics.

Rico Sennrich and Barry Haddow. 2016. Linguistic input features improve neural machine translation. In *Proceedings of the First Conference on Machine Translation: Volume 1, Research Papers*, pages 83–91, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Gabriel Stanovsky, Noah A. Smith, and Luke Zettlemoyer. 2019. Evaluating gender bias in machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1679–1684, Florence, Italy. Association for Computational Linguistics.

Milan Straka and Jana Straková. 2017. Tokenizing, pos tagging, lemmatizing and parsing ud 2.0 with udpipe. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 88–99, Vancouver, Canada. Association for Computational Linguistics.

Yi Chern Tan and L. Elisa Celis. 2019. Assessing social and intersectional biases in contextualized word representations. In H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 13230–13241. Curran Associates, Inc.

Eva Vanmassenhove, Christian Hardmeier, and Andy Way. 2018. Getting gender right in neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3003–3008, Brussels, Belgium. Association for Computational Linguistics.

Eva Vanmassenhove, Dimitar Shterionov, and Andy Way. 2019. Lost in translation: Loss and decay of linguistic richness in machine translation. In *Proceedings of Machine Translation Summit XVII Volume 1: Research Track*, pages 222–232, Dublin, Ireland. European Association for Machine Translation.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2018. Gender bias in coreference resolution: Evaluation and debiasing methods. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 15–20, New Orleans, Louisiana. Association for Computational Linguistics.

Ran Zmigrod, Sabrina J. Mielke, Hanna Wallach, and Ryan Cotterell. 2019. Counterfactual data augmentation for mitigating gender stereotypes in languages with rich morphology. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1651–1661, Florence, Italy. Association for Computational Linguistics.

Document-aligned Japanese-English Conversation Parallel Corpus

Matīss Rikters, Ryokan Ri, Tong Li and Toshiaki Nakazawa
The University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo, Japan
{matiss, li0123, litong, nakazawa}@logos.t.u-tokyo.ac.jp

Abstract

Sentence-level (SL) machine translation (MT) has reached acceptable quality for many high-resourced languages, but not document-level (DL) MT, which is difficult to 1) train with little amount of DL data; and 2) evaluate, as the main methods and data sets focus on SL evaluation. To address the first issue, we present a document-aligned Japanese-English conversation corpus, including balanced, high-quality business conversation data for tuning and testing. As for the second issue, we manually identify the main areas where SL MT fails to produce adequate translations in lack of context. We then create an evaluation set where these phenomena are annotated to alleviate automatic evaluation of DL systems. We train MT models using our corpus to demonstrate how using context leads to improvements.

1 Introduction

The quality of machine translation (MT) for written text and monologue has vastly improved due to the increased amount of available parallel corpora and recent neural network technologies. However, there is much room for improvement in the context of dialogue or conversation translation. One typical case is the translation from a pro-drop language to a non-pro-drop language where correct pronouns must be supplemented according to the context. The omission of the pronouns occurs more frequently in spoken language than written language. Recently, context-aware MT models attract attention from many researchers (Tiedemann and Scherrer, 2017; Voita et al., 2019) to solve this kind of problem, however, there are almost no parallel conversation corpora with context information except the rather noisy Open Subtitles corpus (Tiedemann, 2016).

A document and sentence-aligned conversation parallel corpus should be advantageous to push MT research in this field to the next stage. In this paper, we introduce a newly constructed document-aligned (DA) Japanese-English conversation corpus, which contains three sub-corpora: Business Scene Dialogue (BSD (Rikters et al., 2019)), Japanese translation of AMI Meeting Corpus (AMI (McCowan et al., 2005)) and Japanese translation of OntoNotes 5.0 (ON (Weischedel et al., 2011)). The corpus contains multi-person conversations in various situations: business scenes, meetings under specific themes, broadcast conversations and telephone conversations.

We supplement the original BSD part with additional data, increasing its size by almost three times. We also enrich the corpus with speaker information and other useful meta-data, and separate balanced versions of development and evaluation data sets.

2 Related Work

There are many ready-to-use parallel corpora for training MT systems, but most of them are in written languages such as web crawl, patents (Goto et al., 2011), scientific papers (Nakazawa et al., 2016). Even though some parallel corpora are in spoken language, they are mostly monologues (Cettolo et al., 2012; Di Gangi et al., 2019) or contain a lot of noise (Tiedemann, 2016; Pryzant et al., 2018). Most of the MT evaluation campaigns such as WMT[1], WAT[2] adopt the written language, monologue or noisy dialogue parallel corpora for their translation tasks. Among them, there is only one clean, dialogue parallel corpus (Salesky et al., 2018) adopted by IWSLT[3] in the conversational speech translation task.

JParaCrawl (Morishita et al., 2019) is a recently announced large English-Japanese parallel corpus built by crawling the web and aligning parallel

[1] http://www.statmt.org/wmt20/
[2] http://lotus.kuee.kyoto-u.ac.jp/WAT/
[3] http://workshop2019.iwslt.org

Proceedings of the 5th Conference on Machine Translation (WMT), pages 639–645
Online, November 19–20, 2020.

sentences. Its size is impressive, but it is composed of noisy web-crawled data and has many duplicate sentences. Compared to our corpus, JParaCrawl does not have meta-information and is not DA.

Voita et al. (2019) evaluate what modern MT systems struggle with when translating from English into Russian and construct new development and evaluation sets based on human evaluation. The sets target linguistic phenomena - dexis, ellipsis and lexical cohesion. The authors also provide code for a context-aware NMT toolkit that improves upon translating these phenomena. In contrast, our development/evaluation sets contain complete documents of consecutive sentences, not broken up into only the sentences requiring context.

3 Corpus Description

Our corpus consists of 3 sub-corpora, each of which originates from different sources - BSD, AMI, and ON. BSD was newly constructed, while AMI and ON are translations of the existing English versions of these corpora. Detailed statistics of the sub-corpora are provided in Tables 1 and 2. BSD consists of the scenes mentioned in Table 1, ON has only two different scenes - broadcast conversation and telephone conversation, and all documents from AMI belong to the meeting scene. There is no particular taxonomy associated with these scenes. Word counts for the English side of the sub-corpora are shown in Table 3. We do not include word counts for the Japanese side since it uses very little spaces and the final word count depends on tokenisation.

3.1 Construction Process

Business Scene Dialogue

This sub-corpus was entirely newly created without using any pre-existing resources. We asked professional scenario writers to write monolingual scenarios (documents), and then asked professional translators to translate the documents. This process was done for both En ↔ Ja directions to ensure a wide range of lexicons and expressions from both languages.

In conversations, the utterances are often very short and vague, therefore it is possible that they should be translated differently depending on the situations where the conversations are taking place. For example, the Japanese expression 「すみません」 can be translated into several English expressions, such as "Excuse me", "Thank you." or "I'm sorry.", depending on context. By using scene information, it is possible to discriminate the translations, which is hard to do with only the contextual sentences. Furthermore, it may be possible to connect scene information to multi-modal MT, i.e., estimating the scene from visual information. Language used in meetings and presentations is often more formal than general chatting or phone calls. This is especially prevalent in Japanese, which has three distinct levels of politeness in the spoken language. Knowing the scene may be useful for adjusting politeness and formality.

AMI Meeting Parallel Corpus

The original AMI Meeting Corpus is a multi-modal dataset containing 100 hours of meeting recordings in English. The parallel version was constructed by asking professional translators to translate utterances from the original corpus into Japanese. Since the original corpus consists of speech transcripts, the English sentences contain a lot of short utterances (*e.g., "Yeah", "Okay"*) or fillers (*e.g., "Um"*), and these are translated into Japanese as well. Therefore, the AMI sub-corpus contains many duplicates (see Table 6).

OntoNotes 5.0

The original OntoNotes is comprised of various genres of text (news, telephone speech, weblogs, newsgroups, broadcast, talk shows) in three languages (English, Chinese, and Arabic) with additional annotated information - syntax and predicate argument structure, word sense linked to an ontology and coreference. We extracted the English subsets of broadcast conversation (BC) and telephone conversation (Tele), and had professional translators translate them into Japanese.

Development and Evaluation Sets

We provide balanced development and evaluation splits from only the BSD sub-corpus as it is the least noisy part. The documents in these sets are balanced in terms of scenes and original languages. The complete statistics are shown in Table 4.

3.2 Analysis

We extend the analysis conducted for BSD (Rikters et al., 2019) to AMI and ON by investigating contextual information requirements for EN→JA

	JA→EN		EN→JA	
Scene	**Doc.**	**Sent.**	**Doc.**	**Sent.**
face-to-face	535	16,481	458	14,858
phone call	279	8,720	256	7,770
general chatting	233	7,674	239	7,372
meeting	224	7,647	265	8,952
training	37	1,379	47	1,549
presentation	17	499	53	1,899
sum	1,325	42,400	1,318	42,400

Table 1: Document (Doc.) and sentence (Sent.) statistics for the full BSD corpus. JA→EN represents documents written in Japanese and translated into English. EN→JA represents the opposite documents.

Set (Scene)	Documents	Sentences	PA	WK
AMI	171	110,483	4	0
ON (BC)	27	14,354	5	3
ON (Tele)	46	14,075	6	0

Table 2: Statistics for translated version of AMI and ON corpora and errors detected in EN→JA MT.

MT. We randomly sample 200 and 100 sentence pairs from ON and AMI respectively. In the case of ON, 50% of the pairs are from BC and 50% are from Tele. We translate the sentences with Google Translate[4] and check the translations for errors, ignoring fluency or minor grammatical mistakes. Unlike the JA→EN results for BSD, where more than 50% of errors were due to zero anaphora, there are mainly two types of causes for errors we detected in this analysis - phrase ambiguity (PA) and absence of world knowledge (WK). Most of the errors (Table 2) are caused by PA, for which taking context sentences into account can be considered as a possible solution. On the other hand, the documents in ON-BC contain a variety of named entities (e.g., Shia - one of the two main branches of Islam) and abbreviations (e.g., CPC - Communist Party of China). To solve this, either domain-specific training data or additional mechanisms that take WK into account would be required.

3.3 Release and Licensing

The current version of BSD is published on GitHub[5] under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0) license. The English OntoNotes is under the LDC User Agreement for Non-Members and AMI is under Creative Commons Attribution 4.0 license (CC BY 4.0). We plan to release the extended BSD and translations of AMI under the same licenses and are currently negotiating a licensing agreement for the Japanese translations of OntoNotes.

[4]https://translate.google.com/ (November 2019)

[5]https://github.com/tsuruoka-lab/BSD

	Word Count
Development	19,229
Evaluation	19,619
BSD	750,167
AMI	977,467
ON	279,709

Table 3: English side word counts for each of the subcorpora and development/evaluation sets.

4 Machine Translation Experiments

The conversation corpus alone is not big enough to train real-world NMT systems (as demonstrated by Rikters et al. (2019)). However, by increasing the size of the high-quality BSD corpus, we managed to train reasonable NMT systems. The full statistics of our data are shown in Table 6.

4.1 Experiment Setup

For the SL systems, we used Sockeye (Hieber et al., 2017) to train transformer architecture (Vaswani et al., 2017) models with the *transformer-base* parameters until convergence on development data (no improvement on validation perplexity for 10 checkpoints). Each model was trained 3 times on a single Nvidia TITAN V (12GB) GPU. The reported BLEU score results are an average of 3 runs. Training time was about 2 days for models with only our data and about 5 days when using WMT data.

To train our context-aware systems, we experimented with two approaches - sentence concatenation (Tiedemann and Scherrer, 2017) with source side factors (Sennrich and Haddow, 2016) and context-aware decoder (CADec (Voita et al., 2019)). We use the same toolkit and similar parameters as in our SL systems for the former and the CADec toolkit with the default parameters for the latter. For the concatenation context-aware MT, we experimented with two approaches: 1) prepending the previous sentence from the same document, followed by a beginning of sentence tag <*bos*>, to the source sentence; 2) in addition, providing source side factors to specify if a token represents context or the source sentence.

The source side factors that we used for training were either C or S, representing context and

	Development				Evaluation			
	JA→EN		EN→JA		JA→EN		EN→JA	
Scene	**Doc.**	**Sent.**	**Doc.**	**Sent.**	**Doc.**	**Sent.**	**Doc.**	**Sent.**
face-to-face	11	319	12	314	12	381	11	345
phone call	6	176	7	185	6	163	7	212
general chatting	7	223	8	248	7	211	8	212
meeting	7	240	7	219	7	228	7	229
training	1	40	1	23	1	38	1	30
presentation	1	31	1	33	1	31	1	40
sum	33	1029	36	1029	34	1052	35	1052

Table 4: Document (Doc.) and sentence (Sent.) statistics for development and evaluation sets.

the actual source sentence respectively. Examples of source sentences with context and factors are shown in Table 5. The first sentence in the table has no previous context, as it is the first one in the respective document. The second sentence has the first one as context, followed by a beginning of sentence tag <*bos*>, and so on.

Source sentences
<bos> はい、G社お客様相談室のケイトです。
はい、G社お客様相談室のケイトです。<bos> ご用件は？
ご用件は？<bos> もしもし、森といいます。
Source side factors
C S S S S S S S S S S S S S
C C C C C C C C C C C C C C S S S S S
C C C C C C C S S S S S S S S

Table 5: Examples of training data source sentences and the respective source side factors for the concatenated context-aware experiments.

4.2 Results

The results in Table 7 show that decent quality MT models can be trained by using only our corpus. For JA→EN the scores slightly improve by training contextual models (Concatenated and Concatenated + factors), which indicates that there are context-dependent sentences in our evaluation set that benefit from the additional information. We investigate this further by performing human evaluation in Section 5. We did not find a clear reason why models trained with CADec underperformed even our baseline, but one possible explanation could be that it uses three context sentences at once for each sentence and does not overlap them with the previous and next four-sentence lines, which effectively shrinks the training data down to $\frac{1}{4}$th of the original size.

For comparison, we also trained NMT models on WMT20 data (∼13M parallel sentences, excluding *News Commentary v15*; WMT column in Table 7). For these models, we used *newsdev2020* as development data and *News Commentary v15*[6] as evaluation data since *newstest2020* was not yet available at the time and for Japanese *News Commentary v15* was only 1811 sentences long. These models reached 21.14 BLEU for EN→JA and 20.43 BLEU for JA→EN on *News Commentary v15*, but on our evaluation data they under-performed our baselines. This shows that even with 60x the training data these models struggle to translate conversations. By combining all training data the gain over the baselines is only 0.81 - 1.46 BLEU.

Figure 1 shows one example of a Japanese sentence and its translations by the MT systems. There are no pronouns in the source sentence, but there is the noun 「方」, which should be translated into the English pronoun “he”, specifying the person to be the successor to the store. Both systems manage to translate this part correctly, but the baseline generates an additional pronoun in the end instead of “the store”. We observed many similar situations, where the contextual translation still didn’t match the reference and was not perfect, but the selection of pronouns had improved.

5 Human Evaluation

We translated the evaluation set in both directions using our baseline NMT and performed a two step human evaluation similar to Voita et al. (2019). After that, we analysed the remaining sentences to determine which truly require context.

[6]http://www.statmt.org/wmt20/translation-task.html

	Total	Unique
Development	2,051	2,012
Evaluation	2,120	2,070
Training	80,629	74,377
AMI	110,483	75,660
ON	28,429	24,335

Table 6: Total vs. unique sentence pairs of training, development and evaluation BSD data; and AMI and OntoNotes sub-corpora.

	JA→EN	EN→JA
WMT	16.29	12.99
WMT+	18.44	15.33
Baseline	16.98	14.52
CADec	15.31	12.55
Concatenated	17.07	14.15
Concatenated + factors	17.24	14.19

Table 7: MT experiment results in BLEU scores. WMT uses only WMT 2020 data and WMT+ uses WMT 2020 along with our corpus for training. The rest use only our corpus for training.

We used Yahoo! Japan Crowdsourcing[7] for the human evaluation. Evaluation quality was guaranteed using screening questions which were indistinguishable from the real questions. Only those who correctly answered all the screening questions were considered valid evaluators. Each sentence was evaluated by 5 different evaluators.

In the first step, evaluators were asked to mark each sentence individually as OK or Not Good (NG), where OK meant that the general meaning of the original sentence was transferred to the translation, whereas NG meant that the translation is completely unusable. In the second step, we used only the consecutive pairs of sentences, which were both marked as OK in the first step by at least three evaluators, and asked evaluators to mark them as OK if the corresponding translations made sense in context of each other. We calculated the Free-Marginal Kappa (Randolph, 2005) values for the evaluations to measure agreement between evaluators. The results (overall agreement - 67%, Free-marginal kappa - 0.34) show moderate agreement, which is common for crowdsourcing.

5.1 Analysis

As a result of the crowdsourcing campaign (Table 8) we had 228 EN→JA sentence pairs and 208 JA→EN sentence pairs marked as NG in context of each other. We employed two linguistic experts to check the translations along with their respective sources and references to determine their ambiguity and need for additional context. For this step they were also asked to categorise the ambiguity type.

After the final step 9 EN→JA and 43 JA→EN sentence pairs were marked as context-dependent. 38 JA→EN pairs lack pronouns in the source sentence and do not have enough content to produce an unequivocal translation. The other 5 JA→EN pairs contain ambiguous words or phrases, which can be translated differently, depending on the context. For example, 「1組」 can be translated as either “one couple” or “one group”. Similarly in EN→JA, Chinese can refer to language (中国語) or food (中華料理) as shown in Figure 2. Our best contextual models still struggle to translate such ambiguities, while slightly outperforming SL baselines in handling pronouns.

Figure 3 shows example mistranslations of pronouns, where they are omitted (as is often done in the spoken language) on the Japanese side, but expected in the English translation. The contextual MT model does get some of the pronouns right in the first sentence, but perhaps requires longer context for the second one.

Source: おっ、きっとお店の後継者になる方ですね。
Reference: Oh, he must be the successor to the store.
Baseline: Oh, I’m sure he will succeed **you**.
Con.+fact.: Oh, I’m sure he will be the successor to the store.

Figure 1: JA→EN translations of a sentence where the baseline generated an incorrect pronoun, but the concat. + factors system produced a more fitting translation.

Previous Source:	What kind of food should we choose?
Previous Reference:	どういうジャンルにしますか？
Previous MT:	どんな食べ物を選ぶべきか。
Source:	How about **Chinese**?
Reference:	中華料理はどう？
MT:	中国語はどうですか？

Figure 2: EN→JA MT output where *Chinese* is translated into “中国語” (Chinese language) instead of “中華料理” (Chinese food).

6 Conclusion

We presented a document-aligned parallel corpus of English-Japanese conversations intended for training and evaluation of MT systems. We describe the corpus in detail and indicate which linguistic phenomena are challenging for MT. In our

[7] https://crowdsourcing.yahoo.co.jp/

EN→RU		EN→JA		JA→EN	
2000		2051		2051	
NG	OK	NG	OK	NG	OK
140	1649	228	931	208	1174
4%	41%	11%	45%	10%	57%

Table 8: Results of the second step of the crowdsourcing human evaluation compared to EN→RU (Voita et al., 2019). The first row shows sentence pair totals and the last two rows show sentence pairs, where both sentences were marked as "good" individually, evaluated in context of each other as either good or bad pairs.

Prev. Source:	いつ 返事 くれる と 言っ て た？
Prev. Reference:	Did they say when they will get back to you?
Prev. Base.:	when did you say you' d answer me?
Prev. Conc.+f.:	When did they say they will reply?
Source:	来週 早々 に は 、 と 言っ て ま し た 。
Reference:	They said early next week.
Base.:	He told me early next week.
Conc.+f.:	I said it early next week .

Figure 3: JA→EN MT output by baseline (Base.) and concatenated context + factored (Conc.+f.) models of sentences with no pronouns in the source and expected pronouns in the translation.

evaluation set we marked examples, which can have multiple contrasting translations when tackled on the sentence-level. The release will include the full BSD corpus and Japanese translations of AMI and ON along with instructions on how to align them. The original source language, speaker, scene, document, ambiguity type will also be included.

In the future we plan to model speakers and origin languages in MT, as it can help capture broader context (Maruf et al., 2018) and more precise pronoun translations (Vanmassenhove et al., 2018). We are also interested in experimenting with modelling the scene information within the training data to produce more appropriate translations for each of the politeness settings.

Acknowledgements

This work was supported by "Research and Development of Deep Learning Technology for Advanced Multilingual Speech Translation", the Commissioned Research of National Institute of Information and Communications Technology (NICT), JAPAN.

References

Mauro Cettolo, Christian Girardi, and Marcello Federico. 2012. Wit^3: Web inventory of transcribed and translated talks. In *Proceedings of the 16th Conference of the European Association for Machine Translation (EAMT)*, pages 261–268, Trento, Italy.

Mattia Antonino Di Gangi, Roldano Cattoni, Luisa Bentivogli, Matteo Negri, and Marco Turchi. 2019. MuST-C: a Multilingual Speech Translation Corpus. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, Minneapolis, MN, USA.

Isao Goto, Bin Lu, Ka Po Chow, Eiichiro Sumita, and Benjamin Tsou. 2011. Overview of the patent machine translation task at the ntcir-9 workshop. In *Proc. of NTCIR-9 Workshop Meeting*, pages 559–578.

Felix Hieber, Tobias Domhan, Michael Denkowski, David Vilar, Artem Sokolov, Ann Clifton, and Matt Post. 2017. Sockeye: A toolkit for neural machine translation. *ArXiv e-prints*.

Sameen Maruf, André F. T. Martins, and Gholamreza Haffari. 2018. Contextual neural model for translating bilingual multi-speaker conversations. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 101–112, Belgium, Brussels. Association for Computational Linguistics.

Iain McCowan, Jean Carletta, Wessel Kraaij, Simone Ashby, S Bourban, M Flynn, M Guillemot, Thomas Hain, J Kadlec, Vasilis Karaiskos, et al. 2005. The ami meeting corpus. In *Proceedings of the 5th International Conference on Methods and Techniques in Behavioral Research*, volume 88, page 100.

Makoto Morishita, Jun Suzuki, and Masaaki Nagata. 2019. Jparacrawl: A large scale web-based english-japanese parallel corpus. *arXiv preprint arXiv:1911.10668*.

Toshiaki Nakazawa, Manabu Yaguchi, Kiyotaka Uchimoto, Masao Utiyama, Eiichiro Sumita, Sadao Kurohashi, and Hitoshi Isahara. 2016. Aspec: Asian scientific paper excerpt corpus. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 2204–2208, Portorož, Slovenia. European Language Resources Association (ELRA).

Reid Pryzant, Youngjoo Chung, Dan Jurafsky, and Denny Britz. 2018. JESC: Japanese-English Subtitle Corpus. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Justus J Randolph. 2005. Free-marginal multirater kappa (multirater κfree): An alternative to fleiss' fixed-marginal multirater kappa. In *Presented at the*

Joensuu Learning and Instruction Symposium, volume 2005.

Matīss Rikters, Ryokan Ri, Tong Li, and Toshiaki Nakazawa. 2019. Designing the business conversation corpus. In *Proceedings of the 6th Workshop on Asian Translation*, pages 54–61, Hong Kong, China. Association for Computational Linguistics.

Elizabeth Salesky, Susanne Burger, Jan Niehues, and Alex Waibel. 2018. Towards fluent translations from disfluent speech. In *Proceedings of the IEEE Workshop on Spoken Language Technology (SLT)*, Athens, Greece.

Rico Sennrich and Barry Haddow. 2016. Linguistic input features improve neural machine translation. In *Proceedings of the First Conference on Machine Translation: Volume 1, Research Papers*, pages 83–91, Berlin, Germany. Association for Computational Linguistics.

Jörg Tiedemann. 2016. Finding alternative translations in a large corpus of movie subtitle. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 3518–3522, Portorož, Slovenia. European Language Resources Association (ELRA).

Jörg Tiedemann and Yves Scherrer. 2017. Neural machine translation with extended context. In *Proceedings of the Third Workshop on Discourse in Machine Translation*, pages 82–92, Copenhagen, Denmark. Association for Computational Linguistics.

Eva Vanmassenhove, Christian Hardmeier, and Andy Way. 2018. Getting gender right in neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3003–3008, Brussels, Belgium. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Elena Voita, Rico Sennrich, and Ivan Titov. 2019. When a good translation is wrong in context: Context-aware machine translation improves on deixis, ellipsis, and lexical cohesion. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1198–1212, Florence, Italy. Association for Computational Linguistics.

Ralph Weischedel, Eduard Hovy, Mitchell Marcus, Martha Palmer, Robert Belvin, Sameer Pradhan, Lance Ramshaw, and Nianwen Xue. 2011. *OntoNotes: A Large Training Corpus for Enhanced Processing*, chapter 1. Handbook of Natural Language Processing and Machine Translation. Springer.

Findings of the WMT 2020 Shared Task on Automatic Post-Editing

Rajen Chatterjee[(1)], **Markus Freitag**[(2)], **Matteo Negri**[(3)], **Marco Turchi**[(3)]
[(1)] Apple Inc., Cupertino, CA, USA
[(2)] Google Research, Mountain View, CA, USA
[(3)] Fondazione Bruno Kessler, Trento, Italy

Abstract

We present the results of the 6^{th} round of the WMT task on MT Automatic Post-Editing. The task consists in automatically correcting the output of a "black-box" machine translation system by learning from existing human corrections of different sentences. This year, the challenge consisted of fixing the errors present in English Wikipedia pages translated into German and Chinese by state-of-the-art, not domain-adapted neural MT (NMT) systems unknown to participants. Six teams participated in the English-German task, submitting a total of 11 runs. Two teams participated in the English-Chinese task submitting 2 runs each. Due to *i)* the different source/domain of data compared to the past (Wikipedia vs Information Technology), *ii)* the different quality of the initial translations to be corrected and *iii)* the introduction of a new language pair (English-Chinese), this year's results are not directly comparable with last year's round. However, on both language directions, participants' submissions show considerable improvements over the baseline results. On English-German, the top-ranked system improves over the baseline by -11.35 TER and +16.68 BLEU points, while on English-Chinese the improvements are respectively up to -12.13 TER and +14.57 BLEU points. Overall, coherent gains are also highlighted by the outcomes of human evaluation, which confirms the effectiveness of APE to improve MT quality, especially in the new generic domain selected for this year's round.

1 Introduction

MT Automatic Post-Editing (APE) is the task of automatically correcting errors in a machine-translated text. As pointed out by (Chatterjee et al., 2015), from the application point of view, the task is motivated by its possible uses to:

- Improve MT output by exploiting information unavailable to the decoder, or by performing deeper text analysis that is too expensive at the decoding stage;
- Cope with systematic errors of an MT system whose decoding process is not accessible;
- Provide professional translators with improved MT output quality to reduce (human) post-editing effort;
- Adapt the output of a general-purpose MT system to the lexicon/style requested in a specific application domain.

In its 6^{th} round, the APE shared task organized within the WMT Conference on Machine Translation kept the same overall evaluation setting of the previous five rounds. Specifically, the participating systems had to automatically correct the output of an unknown "black box" (neural) MT system by learning from training data containing human revisions of translations produced by the same system.

This year, the task focused on two language pairs: English-German and English-Chinese. The former has been part of the APE evaluation campaigns since 2016 (Bojar et al., 2016), while the latter represents a new entry. A second difference with respect to previous rounds is that, for both language pairs, the source/domain of the data changed from Information Technology (IT) to Wikipedia articles. The third major novelty factor consists in the type of MT systems used to generate the translations to be corrected. Although for the third year in a row the task focused on translations produced by neural MT (NMT) systems, this year these models were not adapted to the target domain.

These radical changes have advantages and disadvantages. On one side, moving away from the

Proceedings of the 5th Conference on Machine Translation (WMT), pages 646–659
Online, November 19–20, 2020.

"narrow" IT domain allowed to test APE technology on the challenging scenario represented by the generic domain of Wikipedia articles. Indeed, as shown in the previous rounds of the task (Chatterjee et al., 2018a, 2019), the high level of repetitiveness of IT data makes this domain easier to model compared to a generic and less repetitive domain, both for MT and APE technology. Moreover, fixing the output of generic NMT models that are not domain-adapted allowed to test APE on lower-quality initial data and verify its potential as a downstream domain adaptation component. On the other side, the disadvantage of changing domain is the reduced possibility to compare results and measure progress across years. Specifically, the lower quality of the original sentences to be corrected (and, in turn, the larger room for improvement left to APE) make the participants' results and the overall technology advancements difficult to analyze in the light of previous rounds.

Six teams participated in the English-German task, submitting eleven runs in total. Two teams participated in the English-Chinese task, submitting two runs each. Similar to last year, all teams developed their systems based on neural technology, which confirms to be the state-of-the-art approach to APE. In most of the cases (see Section 3), participants experimented with the Transformer architecture (Vaswani et al., 2017), either directly or by adapting it to the task. As in previous rounds, their systems exploit information both from the MT output to be corrected and from the corresponding source sentence. This was done either by concatenating the two, as in last year's winning system (Lopes et al., 2019), or by means of multi-source solutions (Zoph and Knight, 2016) successfully explored in the past (Libovický et al., 2016; Chatterjee et al., 2017). Following the recent trends in other NLP areas, the integration of pre-trained BERT-like language models was also considered. Model ensembling and the integration of word/sentence-level quality estimation techniques geared to APE (similar to (Chatterjee et al., 2018b)) were also explored. Finally, also this year participants took advantage of data augmentation techniques, either by creating their own eSCAPE-like corpora (Negri et al., 2018), or by generating synthetic data by adding artificial noise to simulate post-editing errors, or by exploiting external MT candidates as a source of auxiliary information to be concatenated to the input.

The overall evaluation results show significant improvements over the baseline on both the language directions. On **English-German**, where the "*do-nothing*" baseline (see Section 2.3) was 31.56 TER (Snover et al., 2006) and 50.21 BLEU (Papineni et al., 2002), the top-ranked system (20.21 TER, 66.89 BLEU) shows an impressive -11.35 TER reduction, which corresponds to a +16.68 gain in terms of BLEU score. Considering all the submissions, the average gain is -4.89 TER and +6.5 BLEU points, with only one system performing slightly worse than the baseline. Different from last year, where the differences between the top four submissions were not statistically significant, this year we have a clear winner, whose best submission is 6.78 TER points (and 11.12 BLEU points) above the second ranked team. Nevertheless, though possibly favoured by the relatively low baseline results (+14.72 TER and -24.52 BLEU compared to last year), the globally good performance of the participants is a good indicator of overall progress.

The newly proposed **English-Chinese** task is no exception. Here, both participating teams were able to outperform the baseline (59.49 TER and 23.12 BLEU) by a significant margin. The largest gains are up to -12.13 TER and +14.57 BLEU points and, on average for the four submitted runs, they are -8.15 TER and +10.1 BLEU points.

The good results observed with automatic metrics on both the language pairs are confirmed by the human evaluation outcomes. On English-German, for the first time, the top-ranked primary submission is not significantly worse compared to the human post-edited output (suggesting that automatic corrections are indistinguishable from the human ones[1]). All the other systems except one, moreover, are significantly better than the baseline. This also happens for the two primary submissions to the English-Chinese subtask which, however, are both significantly worse than human post-edits.

All in all, the improvements observed on both the language pairs can be most likely ascribed to the lower quality of the initial translations to be corrected. On English-German, the baseline (31.56 TER, 50.21 BLEU) was indeed much lower

[1]A number of factors (related to this year's data and the overall evaluation setting) may have determined this quite surprising finding. Far from claiming to have reached the "human parity" on the APE task, we leave this aspect to future deeper analyses.

than in the past, when the MT systems used were always domain-adapted and hence more competitive. Last year, for instance, the baseline was 16.84 TER (74.73 BLEU), while in none of the previous rounds focusing on this language pair participants had to confront with TER above 25.0 and BLEU below 62.0. On English-Chinese, the baseline was even lower (59.49 TER, 23.12 BLEU), with the lowest scores across all the past six editions of the APE task. On one side, the large gains observed are in line with (and indirectly confirm) previous observations (Bojar et al., 2017; Chatterjee et al., 2018a, 2019) about the difficulty to improve high-quality MT output. Conversely, as we can observe this year, translations of lower quality (like those coming from generic, not domain-adapted models) leave to APE technology a large margin for improvement. On the other side, the observed global gains in both settings motivate further research on APE as a tool for downstream MT adaptation in black-box conditions.

2 Task description

In continuity with all the previous rounds of the APE task, participants were provided with training and development data consisting of (*source*, *target*, *human post-edit*) triplets, and were asked to return automatic post-edits for a test set of unseen (*source*, *target*) pairs.

2.1 Data

For both English-German and English-Chinese, the initial data were selected from English Wikipedia articles and then automatically translated in the two target languages. Although the original English Wikipedia pages were the same, the source sentences eventually used to build the datasets for the two language pairs are different as they were randomly selected.

The released training and development sets consist of (*source*, *target*, *human post-edit*) triplets in which:

- The source (SRC) is a tokenized English sentence;
- The target (TGT) is a tokenized German/Chinese translation of the source, which was produced by a generic, black-box system unknown to participants. For both the languages, translations were obtained from neural MT systems.[2]
- The human post-edit (PE) is a tokenized manually-revised version of the target, which was produced by professional translators.

Test data consists of (*source*, *target*) pairs having similar characteristics of those in the training set. Human post-edits of the test target instances are left apart to measure system performance.

For the **English-German** subtask, the *training*, *development* and *test* sets respectively contain 7,000, 1,000 and 1,000 triplets. Participants were also provided with two additional training resources, which were widely used in the previous rounds. One is the corpus of 4.5 million artificially-generated post-editing triplets described in (Junczys-Dowmunt and Grundkiewicz, 2016). The other resource is the English-German section of the eSCAPE corpus (Negri et al., 2018). It comprises 14.5 million instances, which were artificially generated both via phrase-based and neural translation (7.25 millions each) of the same source sentences.

Also for the **English-Chinese** subtask, the *training*, *development* and *test* sets respectively contain 7,000, 1,000 and 1,000 triplets. For this language pair, however, no additional training resources were provided.

2.1.1 Complexity indicators: repetition rate

Table 1 provides a view of the data from a task difficulty standpoint. For each dataset released in the six rounds of the APE task, it shows the repetition rate of SRC, TGT and PE elements, the TER (Snover et al., 2006) and the BLEU score (Papineni et al., 2002) of the TGT elements (i.e. the original target translations), as well as the TER difference (δ TER) between the top-ranked submission and the task baseline.

The repetition rate measures the repetitiveness inside a text by looking at the rate of non-singleton n-gram types (n=1...4) and combining them using the geometric mean. Larger values indicate a higher text repetitiveness and, as discussed in (Bojar et al., 2016, 2017; Chatterjee et al., 2018a),

[2]Both the NMT systems are based on the standard Transformer architecture (Vaswani et al., 2017) and follow the implementation details described in (Ott et al., 2018). They were trained on publicly available MT datasets including Paracrawl (Esplà et al., 2019) and Europarl (Koehn, 2005), summing up to 23.7M parallel sentences for English-German and 22.6M for English-Chinese.

	2015	2016	2017	2017	2018	2018	2019	2019	2020	2020
Language	En-Es	En-De	En-De	De-En	En-De	En-De	En-De	En-Ru	En-De	En-Zh
Domain	News	IT	IT	Medical	IT	IT	IT	IT	Wiki	Wiki
MT type	PBSMT	PBSMT	PBSMT	PBSMT	PBSMT	NMT	NMT	NMT	NMT	NMT
Rep. Rate SRC	2.905	6.616	7.216	5.225	7.139	7.111	7.111	18.25	0.653	0.81
Rep. Rate TGT	3.312	8.845	9.531	6.841	9.471	9.441	9.441	14.78	0.823	1.27
Rep. Rate PE	3.085	8.245	8.946	6.293	8.934	8.941	8.941	13.24	0.656	1.2
Baseline TER	23.84	24.76	24.48	15.55	24.24	16.84	16.84	16.16	31.56	59.49
Baseline BLEU	n/a	62.11	62.49	79.54	62.99	74.73	74.73	76.20	50.21	23.12
δ TER	+0.31	-3.24	-4,88	-0,26	-6.24	-0.38	-0.78	+0.43	-11.35	-12.13

Table 1: Basic information about the APE shared task data released since 2015: languages, domain, type of MT technology, repetition rate and initial translation quality (TER/BLEU of TGT). The last row (δ TER) indicates, for each evaluation round, the difference in TER between the baseline (i.e. the "*do-nothing*" system) and the top-ranked submission.

suggest a higher chance of learning from the training set correction patterns that are applicable also to the test set.

Over the years, the relation between systems' performance and the repetition rate observed in the data has been analysed in the light of the different values reported in Table 1. Some rounds of the task suggested the hypothesis that large differences in repetitiveness across the datasets give a possible explanation for the variable gains over the baseline achieved by participants. Indeed, in some cases (e.g. in the APE15 task and in the APE17 German-English subtask), low repetition rates seemed to motivate generally low systems' results, while in others (e.g. APE17 English-German subtask) also the opposite was true, with large gains over the baseline associated to high repetition rates. However, the outcomes of other rounds of the task do not support this intuition. In the 2018 round, despite the relatively high repetition rate values observed in the data, evaluation results shown that the influence of data repetitiveness on final APE performance is marginal. The same happened in 2019 (Chatterjee et al., 2019), when the highest repetition rates ever measured in the APE data (English-Russian subtask) were not enough to develop systems able to improve over the baseline.

As discussed in Section 4, this year we are in the opposite situation. On both English-German and English-Chinese, the lowest repetition rates ever measured in the APE data did not prevent participants from achieving considerable gains over the baseline. This confirms that, as hypothesized last year, systems' improvements over the baseline are either scarcely correlated to text repetitiveness or more influenced by other task difficulty indicators.

2.1.2 Complexity indicators: MT quality

Indeed, another important aspect that determines the difficulty of APE is the initial quality of the MT output to be corrected. This can be measured by computing the TER (↓) and BLEU (↑) scores (Baseline TER/BLEU rows in Table 1) using the human post-edits as reference.

As discussed in (Bojar et al., 2017; Chatterjee et al., 2018a, 2019), numeric evidence of a higher quality of the original translations can indicate a smaller room for improvement for APE systems (having, at the same time, less to learn during training and less to correct at test stage). On one side, indeed, training on good (or near-perfect) automatic translations can drastically reduce the number of learned correction patterns. On the other side, testing on similarly good translations can drastically reduce the number of corrections required and the applicability of the learned patterns, thus making the task more difficult.

As observed in the previous APE evaluation rounds, there is a noticeable correlation between translation quality and systems' performance. In 2016 and 2017, on English-German data featuring a similar level of quality (24.76/24.48 TER, 62.11/62.49 BLEU), the top systems achieved significant improvements over the baseline (-3.24 TER and +5.54 BLEU in 2016, -4.88 TER and +7.58 BLEU in 2017). In 2017, on higher quality German-English data (15.55 TER, 79.54 BLEU), the observed gains were much smaller (-0.26 TER, +0.28 BLEU). In 2018, the correction of English-German translations produced by a phrase-based system (24.24 TER, 62.99 BLEU) yielded much larger gains (up to -6.24 TER and +9.53 BLEU) compared to the correction of higher-quality neural translations (16.84 TER, 74.73 BLEU), which resulted in TER/BLEU variations of less than 1.00 point. Similarly, in 2019 the very high translation

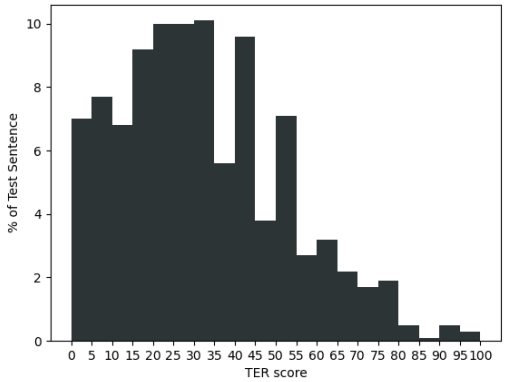


Figure 1: TER distribution in the **English-German** test set

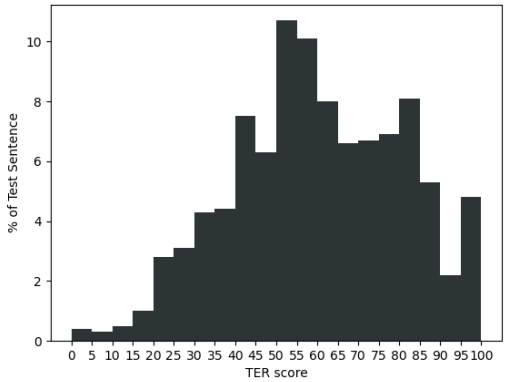


Figure 2: TER distribution in the **English-Chinese** test set

quality featured by strong, domain-adapted neural models made the task rather difficult. On English-German, where the baseline system was again very competitive (16.84 TER, 74.73 BLEU), the largest TER reduction was indeed of 0.78 points (corresponding to a BLEU increase of 1.23). On English-Russian, where the initial MT quality was even higher,[3] (16.16 TER, 76.2 BLEU), the baseline remained unbeaten.

As discussed in Section 4, also this year's results confirm the strict correlation between the quality of the initial translations and the actual potential of APE. Indeed, with baseline TER and BLEU scores significantly lower than in all the other rounds of the task (31.56 TER and 50.21 BLEU for English-German, 59.49 TER and 23.12 BLEU for English-Chinese), almost all participants managed to obtain very large improvements despite the low repetition rates featured by the data.

2.1.3 Complexity indicators: TER distribution

A third complexity indicator considered in previous rounds of the task is the TER distribution (computed against human references) for the translations present in the test sets. What we observed in the past is that harder tasks were typically characterized by TER distributions particularly skewed towards low values. For instance, in 2019 around 50% of the English-German and 63.5% of the English-Russian test items had a TER between 0 and 10, the latter subtask being considerably more difficult than the former (recall that, on English-Russian, none of the participants was able to beat the baseline). Indeed, the higher the proportion of (near-)perfect test instances (i.e. items with $0<\text{TER}<10$, which hence require few edits or no corrections at all), the higher the probability that APE systems will perform unnecessary corrections that will be penalized by automatic evaluation metrics.

On the contrary, less skewed distributions can be expected to be easier to handle as they give to automatic systems a larger room for improvement. In the lack of more focused analyses on this aspect, we can hypothesize that, in ideal conditions from the APE standpoint, the peak of the distribution would be observed for "post-editable" translations containing enough errors that leave some margin for focused corrections, but not too many errors to be so unintelligible to require a whole re-translation from scratch.[4]

As shown in Figures 1 and 2, the TER distributions in the two test sets released this year is quite different from previous rounds and actually reflects a more balanced situation. For English-German, about 55% of the samples falls in the 15-45 TER interval, with no more $\sim 7\%$ of the items being perfect (i.e. TER=0). For English-Chinese, for which the overall MT quality is significantly lower (as shown by the worse baseline results reported in Table 1), the vast majority of the samples falls in the 40-85 interval, with less than 1% of the

[3] Note that the higher quality of the initial translations added up to the higher difficulty of dealing with a morphologically-rich target language like Russian. The two aspects are clearly tightly connected and disentangling them would require further analysis. Nonetheless, regarding the correlation between MT quality and final results, also this subtask was not an exception compared to the other settings summarized in Table 1.

[4] For instance, based on the empirical findings reported in (Turchi et al., 2013, 2014), TER=0.4 is the threshold that, for human post-editors, separates the "post-editable" translations from those that require complete rewriting from scratch.

ID	Participating team
MinD	Alibaba Group, Hangzhou, China (Wang et al., 2020)
BeringLab	Bering Lab, Republic of Korea (Lee, 2020)
HW-TSC	Huawei Translation Services Center & East China Normal University, China (Yang et al., 2020)
KAIST	Korea Advanced Institute of Science & Technology, Republic of Korea
POSTECH	Pohang University of Science and Technology, Republic of Korea (Lee et al., 2020b)
POSTECH-ETRI	Pohang University & Electronics and Telecomm. Res. Inst., Republic of Korea (Lee et al., 2020a)

Table 2: Participants in the WMT20 Automatic Post-Editing task.

items being perfect.

In the light of previous years' observations, both the subtasks hence seem to be easier to handle. As discussed in Section 4, also this year's evaluation results confirm the strict correlation between the quality of the initial translations, the distribution of TER scores across the test items, and the actual potential of APE.

2.2 Evaluation metrics

System performance was evaluated both by means of automatic metrics and manually. Automatic metrics were used to compute the distance between *automatic* and *human* post-edits of the machine-translated sentences present in the test sets. To this aim, TER and BLEU (case-sensitive) were respectively used as primary and secondary evaluation metrics. Systems were ranked based on the average TER calculated on the test set by using the TERcom[5] software: lower average TER scores correspond to higher ranks. BLEU was computed using the multi-bleu.perl package[6] available in MOSES. The evaluation results computed in terms of automatic metrics are presented and discussed in Section 4).

Manual evaluation was conducted via source-based direct human assessment (Graham et al., 2013; Cettolo et al., 2017; Bojar et al., 2018). Details are discussed in Section 6.

2.3 Baseline

In continuity with the previous rounds, the official baseline results were the TER and BLEU scores calculated by comparing the raw MT output with human post-edits. In practice, the baseline APE system is a "*do-nothing*" system that leaves all the test targets unmodified. Baseline results, the same shown in Table 1, are also reported in Tables 3 and 4 for comparison with participants' submissions.[7]

For each submitted run, the statistical significance of performance differences with respect to the baseline was calculated with the bootstrap test (Koehn, 2004).

3 Participants

Six teams submitted a total of 11 runs for the English-German subtask. Two of them participated also in the English-Chinese subtask by submitting 2 runs each. Participants are listed in Table 2, and a short description of their systems is provided in the following.

Alibaba Group (*MinD*). Alibaba participated only in the English-German subtask. Their submission introduces a cross-lingual Bert-like conditional model with a "memory-encoder", which can capture the semantic information of machine translations conditional on the source sentences (Fan et al., 2019). The system consists of three parts, namely: *i)* a general Transformer encoder to encode the source sentences, *ii)* a Transformer decoder without future mask adapted to a memory-encoder to encode machine translations with cross attention to the source encoder, and *iii)* a multi-source Transformer decoder to generate the automatic post-editing results with cross attentions to both the encoders. In addition, data augmentation, corpus filtering and imitation learning strategies are exploited to overcome the scarcity of real APE data and to further improve model's performance, together with model ensembling and conservativeness penalty strategies inspired by (Lopes et al., 2019).

[5] http://www.cs.umd.edu/~snover/tercom/

[6] https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/multi-bleu.perl

[7] In addition to the *do-nothing* baseline, in the first three rounds of the task we also compared systems' performance with a re-implementation of the phrase-based approach firstly proposed by Simard et al. (2007), which represented the common backbone of APE systems before the spread of neural solutions. As shown in (Bojar et al., 2016, 2017), the steady progress of neural APE technology has made the phrase-based solution not competitive with current methods reducing the importance of having it as an additional term of comparison. Since 2018, we hence opted for considering only one baseline.

Bering Lab (*BerlingLab*). Bering Lab participated only in the English-German subtask. Their system relies on a Transformer architecture, in which the encoder takes in input a concatenation of the source and MT sentences to generate a cross-lingual representation to be passed to the decoder. Additionally, they explored methods to improve APE performance through word-level and sentence-level quality estimation (QE). Based on word-level QE, they mask incorrect or missing words in the PE output. Then, the most probable word for each masked token is predicted using XLM-RoBERTa (Conneau et al., 2020), which is fine-tuned based on the translation language modeling (TLM) objective (Conneau and Lample, 2019). Finally, they propose an output selection mechanism based on sentence-level QE to prevent over-correction. To this aim, they select the sentence with the lowest predicted HTER among the PE outputs and the original MT sentence as the final output. For data augmentation, they use a parallel corpus to train an NMT model and generate artificial triplets, following the ideas from (Negri et al., 2018).

Huawei (*HW-TSC*). Huawei participated both in the English-German and English-Chinese subtasks. Their system basically follows the architecture of last year's winning system (Lopes et al., 2019), where *src* and *mt* sentences are concatenated as input to the encoder, and the decoder is used for decoding the *pe* sentence. However, there are several differences with respect to (Lopes et al., 2019). First, instead of using a pre-trained BERT model, the system relies on a Transformer NMT model (implemented with fairseq (Ott et al., 2019)), pre-trained on the WMT19 news translation corpora. Second, the model integrates bottleneck adapter layers to prevent from over-fitting. Third, external MT candidates (from Google Translate) are exploited as a source of auxiliary information. This results in a longer input sequence composed of (*src*, *mt*, *auxiliary_mt*) triplets. Due to the domain change introduced this year, system's training does not exploit the supplied additional corpora for data augmentation. Finally, the system does not include methods to prevent over-correction, such as the penalty mentioned in (Lopes et al., 2019).

POSTECH (*POSTECH_TERNoise*). This team participated only in the English-German subtask. They mainly focused on increasing the size of the APE data to overcome the scarcity of training samples available. They first introduced a noising module simulating the four types of post-editing errors: insertion, deletion, substitution and shifting. This noising module implants the simulated errors into the target text of the parallel corpora, so to exploit a synthetic MT output during the training phase. The quantity of noise is determined by using the TER distribution of the official training set. They then applied the same generation method proposed in (Negri et al., 2018), so to create a synthetic APE corpus to be used as additional training data. For this data construction process, they used the parallel corpora and the NMT model released for the WMT20 Quality Estimation shared task. As APE model, they chose the sequential model proposed in (Lee et al., 2019), applying some minor modifications to increase the training efficiency. They submitted two ensemble models. Their primary submission (TERNoise-Ops-Ens8) is an ensemble of eight runs. It was obtained by first selecting the top-5 runs having the lowest TER on the development set, for three individual weight initializations. Out of them, they then selected the top-2 runs showing most frequent corrections for each of the four edit operations to form the ensemble. The contrastive submission (TERNoise-nFold-Ens8) is an ensemble of eight runs obtained from models trained/validated in a 4-fold setting on the integration of training data and development data, aiming at the generalization to unseen data. Then, the top-2 runs for each fold were selected to form the ensemble.

POSTECH-ETRI (*POSTECH-ETRI*). This team participated both in the English-German and English-Chinese subtasks. Their models focus on adapting to the APE task XLM (Conneau and Lample, 2019), which can learn joint representations from two languages. Rather than using the open model published on the XLM github page[8] trained on 15 languages, they built new MLM+TLM models that are trained on datasets consisting of only the source and target languages for both language pairs (English-German and English-Chinese). Their model architecture is an extension of Transformer, in which the encoder is initialized with the weights of the pre-trained

[8] https://github.com/facebookresearch/XLM

		TER	BLEU
en-de	HW-TSC_DIRECT_CONTRASTIVE.pe	20.21	66.89
	HW-TSC_CONCAT_PRIMARY.pe	20.52	66.16
	MinD-mem_enc_dec_post-CONTRASTIVE	26.99	55.77
	POSTECH-ETRI_XLM-Top4Ens_CONTRASTIVE	27.02	56.37
	MinD-mem_enc_dec-PRIMARY	27.03	55.58
	POSTECH-ETRI_XLM-Top3Ens_PRIMARY	27.37	55.83
	BeringLab_model1_PRIMARY	27.61	54.71
	BeringLab_model2_CONTRASTIVE	27.96	54.60
	POSTECH_TERNoise-nFold-Ens8_CONTRASTIVE	28.22	54.51
	POSTECH_TERNoise-Ops-Ens8_PRIMARY	28.41	54.22
	Baseline	31.56	50.21
	KAISTxPAPAGO_EMT_PRIMARY	32.00	49.21

Table 3: Results for the WMT20 APE **English-German** – average TER (↓), BLEU score (↑).

		TER	BLEU
en-zh	HW-TSC_CONCAT_PRIMARY.pe	47.36	37.69
	HW-TSC_DIRECT_CONTRASTIVE.pe	48.01	37.32
	POSTECH-ETRI_XLM-Top3Ens_PRIMARY	54.92	28.90
	POSTECH-ETRI_XLM-Top4Ens_CONTRASTIVE	55.08	28.97
	Baseline	59.49	23.12

Table 4: Results for the WMT20 APE **English-Chinese** – average TER (↓), BLEU score (↑).

XLM, receiving the concatenation of the two input sentences. The decoder is also initialized in a similar manner as the encoder, while multi-head attention layers are random-initialized. At the APE training stage, in addition to the WMT20 official dataset, they generated new synthetic triplets, following the same method used to build eSCAPE (Negri et al., 2018). They used the NMT model provided by the WMT20 quality estimation shared task to generate new synthetic APE triplets by translating the parallel corpus provided by the same task. Finally, to generate their final submissions, they built an ensemble of multiple models.

4 Results

Participants' results are shown in Tables 3 (English-German) and 4 (English-Chinese). The submitted runs are ranked based on the average TER (case-sensitive) computed using human post-edits of the MT segments as reference, which is the APE task primary evaluation metric. The two tables also report the BLEU score computed using human post-edits, which represents our secondary evaluation metric.

Similar to last year, also in this round the primary and secondary evaluation metric produce rankings that are only slightly different from each other.[9] In spite of these minor difference, for both both languages we have a clear separation between the two top-ranked submissions (by the same team) and the other submitted runs.

On **English-German**, the best results (20.21 TER, 66.89 BLEU) respectively outperform the baseline by -11.35 TER and +16.68 BLEU points, the second-best scores being lower by less than 1 point for both the metrics. All the other runs but the last are quite close to each other, being concentrated respectively in a 1.42 TER and 1.55 BLEU points interval.

On **English-Chinese**, the best results (47.36 TER, 37.69 BLEU) respectively outperform the baseline by -12.13 TER and +14.57 BLEU points. Also in this case, the second-best run is below the top-ranked one by less than 1 point for both the metrics, while the third and fourth submissions are close to each other (the difference is less than 0.2 points for both metrics).

All in all, these results indicate that:

- Operating with lower-quality output produced by generic (i.e. not domain-adapted) NMT systems run on a broad "domain" like Wikipedia texts (as opposed to the narrow domains of information technology or medical) leaves considerable room for improvement to state-of-the-art APE models. Looking at the baseline scores and the δTER values shown

[9] For English-German, the third and fourth-ranked submissions in terms of TER are switched in terms of BLEU, as well as the fifth and the sixth. For English-Chinese, this happens for the third and fourth-ranked submissions. The correlation between the ranks obtained by the two metrics is however very high, and in both cases above 0.99.

Systems	Modified	Improved	Deteriorated	Prec.
HW-TSC_DIRECT_CONTRASTIVE.pe	905 (90.5%)	625 (69.06%)	177 (19.56%)	0.69
HW-TSC_CONCAT_PRIMARY.pe	908 (90.8%)	618 (68.06%)	183 (20.15%)	0.68
MinD-mem_enc_dec_post-CONTRASTIVE	662 (66.2%)	397 (59.97%)	148 (22.36%)	0.60
POSTECH-ETRI_XLM-Top4Ens_CONTRASTIVE	771 (77.1%)	438 (56.81%)	199 (25.81%)	0.57
MinD-mem_enc_dec-PRIMARY	665 (66.5%)	401 (60.30%)	144 (21.65%)	0.60
POSTECH-ETRI_XLM-Top3Ens_PRIMARY	778 (77.8%)	423 (54.37%)	207 (26.61%)	0.54
BeringLab_model1_PRIMARY	708 (70.8%)	380 (53.67%)	157 (22.18%)	0.54
BeringLab_model2_CONTRASTIVE	421 (42.1%)	279 (66.27%)	72 (17.10%)	0.66
POSTECH_TERNoise-nFold-Ens8_CONTRASTIVE	535 (53.5%)	306 (57.20%)	108 (20.19%)	0.57
POSTECH_TERNoise-Ops-Ens8_PRIMARY	536 (53.6%)	309 (57.65%)	112 (20.90%)	0.58
KAISTxPAPAGO_EMT_PRIMARY	724 (72.4%)	267 (36.88%)	314 (43.37%)	0.37
Average	69.2	58.2	23.6	0.58

Table 5: Number (raw and proportion) of test sentences modified, improved and deteriorated by each run submitted to the APE 2020 **English-German** subtask. The "Prec." column shows systems' precision as the ratio between the number of improved sentences and the total number of modified instances.

Systems	Modified	Improved	Deteriorated	Prec.
HW-TSC_CONCAT_PRIMARY.pe	997 (99.7%)	673 (67.50%)	227 (22.77%)	0.68
HW-TSC_DIRECT_CONTRASTIVE.pe	995 (99.5%)	671 (67.44%)	223 (22.41%)	0.67
POSTECH-ETRI_XLM-Top3Ens_PRIMARY	968 (96.8%)	566 (58.47%)	265 (27.38%)	0.58
POSTECH-ETRI_XLM-Top4Ens_CONTRASTIVE	959 (95.9%)	551 (57.46%)	255 (26.59%)	0.57
Average	97.975	62.72	24.79	0.63

Table 6: Number (raw and proportion) of test sentences modified, improved and deteriorated by each run submitted to the APE 2020 **English-Chinese** subtask. The "Prec." column shows systems' precision as the ratio between the number of improved sentences and the total number of modified instances.

in Table 1, we can observe that the largest improvements over the baseline were obtained this year on the lowest-quality translations.

- Operating with data featuring low repetition rates does not necessarily prevent from obtaining significant MT quality improvements. Looking at the δTER and the repetition rate values shown in Table 1, we can observe that the lowest data repetitiveness observed this year did not prevent from observing, at the same time, the largest gains over the baseline.

- Operating with data featuring variable quality, with a distribution of the instances that is not too peaked towards high-quality translations, sets ideal conditions for APE. Looking at the δTER and the TER distributions shown in Figures 1 and 2, we can observe that the largest improvements over the baseline achieved this year are also related to a quality distribution that is more uniformly spread around central values of the 0-100 TER interval.

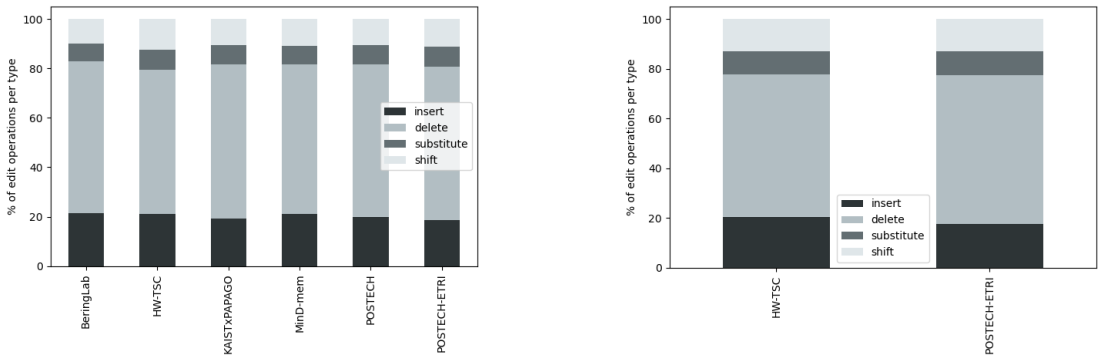
5 System/performance analysis

As a complement to global TER/BLEU scores, also this year we performed a more fine-grained analysis of the changes made by each system to the test instances.

5.1 Macro indicators: modified, improved and deteriorated sentences

Tables 5 and 6 show, for each run submitted to the two subtasks, the number of modified, improved and deteriorated sentences, as well as the overall system's precision (i.e. the proportion of improved sentences out of the total number of modified instances). It's worth noting that, as in the previous rounds and in both the settings, the number of sentences modified by each system is higher than the sum of the improved and the deteriorated ones. This difference is represented by modified sentences for which the corrections do not yield TER variations. This grey area, for which quality improvement/degradation can not be automatically assessed, contributes to motivate the human evaluation discussed in Section 6.

As shown in Table 5, on **English-German** the amount of sentences modified by the participants varies from the very high values of the top two submissions (above 90.0%) to the lower scores of the runs placed below them in the ranking (between 42.1% and 77.8%). However, in all the cases the overall number of modified sentences (69.2% on average) is considerably larger than what we observed in the 2019 round (23.53% on average, ranging from 4.01% to 39.1%). This difference can be ascribed to the different nature of the data that, as previously discussed, this year



(a) (b)

Figure 3: System behaviour (primary submissions) for **English-German** (a) and **English-Chinese** (b) – TER(MT, APE)

featured lower MT quality, combined with a distribution that is less skewed towards low TER values. In particular, while last year about 30.0% of the test instances were to be considered as "perfect", this year the proportion of test instances with 0≤TER≤5 is about 7.0%. In light of this, compared to last year, the participants modified a number of test instances that is much closer to the target percentage of sentences to be modified (about 93.0%, i.e. those having TER>0). As one can expect, besides systems' aggressiveness, final performance highly depends also on their precision in applying corrections. The last column of Table 5 shows systems' precision (Prec.) as the ratio between the number of improved sentences and the total number of modified sentences. As can be seen from the table, the two top-ranked submissions are not only the most aggressive (more than 90% modified sentences) but also the most precise ones (precision above 0.68). Overall, all runs but one have a precision above 0.5, with an average value of 0.58 that is larger than the values observed on the same language (but different evaluation conditions) in 2019 (0.46) and in 2018 (0.34). As a consequence, the percentage of deteriorated sentences out of the total amount of modified test items shows a significant drop with respect to the last two rounds of the task. On average, a quality decrease is observed for 23.6% of the test items (it was 47.85% in 2018 and 35.11% in 2019).

As shown in Table 6, on **English-Chinese** we observe similar trends. The four submitted runs are all characterized by a high percentage of modified sentences (97.97% on average) and a very high precision (0.63 on average). This can be explained by the large room for improvement available to APE on this language pair, due to the low MT baseline (59.49 TER, 23.12 BLEU) and to the small number of "perfect" translations (as shown in Figure 2, less than 0.5% of the test items have a 0≤TER≤5).

5.2 Micro indicators: edit operations

In the previous rounds of the APE task, possible differences in the way systems corrected the test set instances were analyzed by looking at the distribution of the edit operations done by each system (insertions, deletions, substitutions and shifts). Such distribution was obtained by computing the TER between the original MT output and the output of each system taken as reference (only for the primary submissions). This analysis has been performed also this year but it turned out to be scarcely informative, as shown in Figure 3. For both the subtasks, the differences in system's behaviour are indeed barely visible. All the submitted runs are characterized by a large number of deletions (on average, 61.11% for English-German and 58.54% for English-Chinese), followed by the insertions (respectively, 20.17% and 19.01%), the shifts (10.98% and 12.98%) and finally the substitutions (7.74 and 9.48). These distributions differ from what we observed in the past. Especially in the last two rounds of the APE task, the largest proportion of edit operations were indeed substitutions (for English-German neural translations, they were 53.6% in 2019 and 53.5% in 2018). Also this difference can be explained by the lower quality of this year's initial translations. In the previous rounds, the generally high fluency of domain-adapted neural MT systems induced the trained APE models to perform light changes, mainly with isolated word substitutions oriented to improve lexical choice. This year, the change of domain and the use of generic models that were not domain-adapted resulted in more

aggressive structural modifications, where lexical changes represent the minority of edit operations.

6 Human evaluation

In order to complement the automatic evaluation of APE submissions, manual evaluation of the primary systems submitted (seven for English-German, three for English-Chinese) was conducted. In this section, we present the evaluation procedure, as well as the results obtained.

6.1 Evaluation procedure

We evaluated the overall quality of the MT and PE output using source-based direct assessment (Graham et al., 2013; Cettolo et al., 2017; Bojar et al., 2018). We used the same instructions that are used in the News Translation track of WMT2020. We hired 25 professional linguists for English-German and 25 professional linguists for English-Chinese. All involved linguists were either native speaker in German or Chinese.

We acquired only a single rating per sentence as we found that professional linguists were more reliable than crowd workers (Toral, 2020). For adequacy, we asked annotators to assess the semantic similarity between the source and a candidate text, labelled as "source text" and "candidate translation", respectively. The annotation interface implements a slider widget to encode perceived similarity as a value between 0 and 100. Note that the exact value is hidden from the human, and can only be guessed based on the positioning of the slider. Candidates are displayed in random order, so to prevent biased assessments.

For our human evaluation campaign, we also include the human post-edits (test.pe) and the unedited, MT output (test.mt). We expect human post-editing to be of higher quality than the output from APE submissions, which in turn should outperform the unedited MT output. We run human evaluation for all primary submissions, the MT output and the human post-edited output.

6.1.1 English→German

Human evaluation results for English-German are summarized in Table 7. The human post-edited output *test.pe* scores best, while the APE output *HW-TSC_CONCAT.pe* is not significantly worse compared to the human post-edited output. Consequently, and rather surprisingly, human and automatic corrections for this language pair seem to be indistinguishable to our evaluators. This interesting finding can be motivated by a number of reasons (the type/quality/quantity of data, the size of the sample, the number of collected judgements) that suggest to avoid exaggerated claims about a reached human parity. Nonetheless, we take it as indicator of a steady progress of APE research. Interestingly, 5 out of 6 APE submissions perform significantly better than the original MT output *test.mt*, demonstrating that APE can be used to improve machine translation output even for high-resource language settings like English-German, as already shown by Freitag et al. (2019). These findings are different from last year's APE task (Chatterjee et al., 2019) where none of the English-German APE submissions was significantly better than the raw MT output.

	Avg	Avg z
test.pe	83.5	0.298
HW-TSC_CONCAT.pe	82.2	0.260
POSTECH-ETRI_XLM-Top3Ens	77.3	0.031
MinD-mem_enc_dec	76.2	-0.008
POSTECH_TERNoise-Ops-Ens8	75.8	-0.037
BeringLab_model1	74.3	-0.098
test.mt	71.5	-0.194
KAISTxPAPAGO_EMT	71.0	-0.252

Table 7: Results for the WMT20 APE **English-German – human evaluation**. Systems ordered by DA score z-score; systems within a cluster are considered tied; lines indicate clusters according to Wilcoxon rank-sum test $p < 0.05$.

	Avg	Avg z
test.pe	86.3	0.363
HW-TSC_CONCAT.pe	77.2	-0.063
POSTECH-ETRI_XLM-Top3Ens	77.0	-0.079
test.mt	74.0	-0.221

Table 8: Results for the WMT20 APE **English-Chinese – human evaluation**. Systems ordered by DA score z-score; systems within a cluster are considered tied; lines indicate clusters according to Wilcoxon rank-sum test $p < 0.05$.

6.1.2 English→Chinese

Human evaluation results for English-Chinese are summarized in Table 8. In this case, the human post-edited output does perform significantly better than the two primary submissions. Similar to the English-German task, both APE submissions perform significantly better than the original MT output *test.mt*. Nevertheless, both submissions

perform very similarly, and both submissions can be seen as similar quality.

7 Conclusion

We presented the results from the 6^{th} shared task on Automatic Post-Editing at WMT. This year, we proposed two subtasks in which the MT output to be corrected was respectively generated by English-German and English-Chinese neural systems unknown to the participants. The latter language pair represents a new entry for the task, which previously focused on Spanish (in 2015), German (since 2016) and Russian (in 2019) as target languages. The other major novelty factors are that: *i)* both the subtasks dealt with data drawn from the "generic" domain of Wikipedia articles, and *ii)* the NMT systems used to generate the translations were not domain-adapted. As a consequence, participants had to confront with lower quality translations that left to APE large room for improvement.

Six teams participated in the English-German task, with a total of 11 submitted runs, while two teams participated in the English-Chinese task submitting two runs each. Their results computed with automatic metrics (TER and BLEU) revealed significant gains over the "*do-nothing*" baseline. On English-German, the top-ranked system improved over the baseline by -11.35 TER and +16.68 BLEU points, and the average improvements were the largest ones observed over the years (-4.89 TER, +6.5 BLEU). On English-Chinese the improvements of the top-ranked system are respectively -12.13 TER and +14.57 BLEU points, with average gains of (-8.15 TER and +10.1 BLEU). Our human evaluation confirmed that on both the language pairs, almost all the primary submissions are significantly better than the baseline. On English-German, the improvement is up to the point that the quality of the automatic corrections produced by the top-ranked primary submissions is substantially on par with human corrections.

All in all, these results confirm the effectiveness of APE to improve MT output in black-box conditions, especially when the adaptation of generic systems to a new "domain" is required.

Acknowledgments

We would like to thank Apple and Google Research for their support and sponsorship in organizing the 2020 APE shared task.

References

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Raphael Rubino, Lucia Specia, and Marco Turchi. 2017. Findings of the 2017 Conference on Machine Translation (WMT17). In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 169–214, Copenhagen, Denmark. Association for Computational Linguistics.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurelie Neveol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. 2016. Findings of the 2016 conference on machine translation. In *Proceedings of the First Conference on Machine Translation*, pages 131–198, Berlin, Germany. Association for Computational Linguistics.

Ondřej Bojar, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, and Christof Monz. 2018. Findings of the 2018 conference on machine translation (wmt18). In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 272–307, Belgium, Brussels. Association for Computational Linguistics.

Mauro Cettolo, Marcello Federico, Luisa Bentivogli, Jan Niehues, Sebastian Stüker, Katsuhito Sudoh, Koichiro Yoshino, and Christian Federmann. 2017. Overview of the iwslt 2017 evaluation campaign. In *Proc. of IWSLT*, Tokyo, Japan.

Rajen Chatterjee, M. Amin Farajian, Matteo Negri, Marco Turchi, Ankit Srivastava, and Santanu Pal. 2017. Multi-source Neural Automatic Post-Editing: FBK's Participation in the WMT 2017 APE Shared Task. In *Proceedings of the Second Conference on Machine Translation*, pages 630–638. Association for Computational Linguistics.

Rajen Chatterjee, Christian Federmann, Matteo Negri, and Marco Turchi. 2019. Findings of the WMT 2019 shared task on automatic post-editing. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 11–28, Florence, Italy. Association for Computational Linguistics.

Rajen Chatterjee, Matteo Negri, Raphael Rubino, and Marco Turchi. 2018a. Findings of the WMT 2018

shared task on automatic post-editing. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 710–725, Belgium, Brussels. Association for Computational Linguistics.

Rajen Chatterjee, Matteo Negri, Marco Turchi, Frédéric Blain, and Lucia Specia. 2018b. Combining Quality Estimation and Automatic Post-editing to Enhance Machine Translation Output. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Papers)*, pages 26–38, Boston, MA. Association for Machine Translation in the Americas.

Rajen Chatterjee, Marion Weller, Matteo Negri, and Marco Turchi. 2015. Exploring the Planet of the APEs: a Comparative Study of State-of-the-art Methods for MT Automatic Post-Editing. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 156–161, Beijing, China. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 8440–8451. Association for Computational Linguistics.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 7057–7067.

Miquel Esplà, Mikel Forcada, Gema Ramírez-Sánchez, and Hieu Hoang. 2019. ParaCrawl: Web-scale parallel corpora for the languages of the EU. In *Proceedings of Machine Translation Summit XVII Volume 2: Translator, Project and User Tracks*, pages 118–119, Dublin, Ireland. European Association for Machine Translation.

Kai Fan, Jiayi Wang, Bo Li, Boxing Chen, and N. Ge. 2019. Neural zero-inflated quality estimation model for automatic speech recognition system. *ArXiv*, abs/1910.01289.

Markus Freitag, Isaac Caswell, and Scott Roy. 2019. APE at Scale and Its Implications on MT Evaluation Biases. In *Proceedings of the Fourth Conference on Machine Translation*, pages 34–44, Florence, Italy. Association for Computational Linguistics.

Yvette Graham, Timothy Baldwin, Alistair Moffat, and Justin Zobel. 2013. Continuous Measurement Scales in Human Evaluation of Machine Translation. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 33–41, Sofia, Bulgaria. Association for Computational Linguistics.

Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2016. Log-linear Combinations of Monolingual and Bilingual Neural Machine Translation Models for Automatic Post-Editing. In *Proceedings of the First Conference on Machine Translation*, pages 751–758, Berlin, Germany.

Philipp Koehn. 2004. Statistical Significance Tests for Machine Translation Evaluation. In *Proceedings of EMNLP 2004*, pages 388–395, Barcelona, Spain.

Philipp Koehn. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Proceedings of the Tenth Machine Translation Summit (MT Summit X)*, pages 79–86, Phuket, Thailand.

Dongjun Lee. 2020. Cross-Lingual Transformers for Neural Automatic Post-Editing. In *Proceedings of the Fifth Conference on Machine Translation*, Online.

Jihyung Lee, WonKee Lee, Jaehun Shin, Baikjin Jung, Young-Kil Kim, and Jong-Hyeok Lee. 2020a. POSTECH-ETRI's Submission to the WMT2020 APE Shared Task: Automatic Post-Editing with Cross-lingual Language Model. In *Proceedings of the Fifth Conference on Machine Translation*, Online.

WonKee Lee, Jaehun Shin, Baikjin Jung, Jihyung Lee, and Jong-Hyeok Lee. 2020b. Noising Scheme for Data Augmentation in Automatic Post-Editing. In *Proceedings of the Fifth Conference on Machine Translation*, Online.

WonKee Lee, Jaehun Shin, and Jong-Hyeok Lee. 2019. Transformer-based automatic post-editing model with joint encoder and multi-source attention of decoder. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 112–117, Florence, Italy. Association for Computational Linguistics.

Jindřich Libovický, Jindřich Helcl, Marek Tlustý, Ondřej Bojar, and Pavel Pecina. 2016. CUNI at Post-editing and Multimodal Translation Tasks. In *Proceedings of the 11th Workshop on Statistical Machine Translation (WMT)*.

António V. Lopes, M. Amin Farajian, Gonçalo M. Correia, Jonay Trénous, and André F. T. Martins. 2019. Unbabel's submission to the WMT2019 APE shared task: BERT-based encoder-decoder for automatic post-editing. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 118–123, Florence, Italy. Association for Computational Linguistics.

Matteo Negri, Marco Turchi, Rajen Chatterjee, and Nicola Bertoldi. 2018. eSCAPE: a Large-scale Synthetic Corpus for Automatic Post-Editing. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. 2018. Scaling neural machine translation. In *Proceedings of the Third Conference on Machine Translation (WMT)*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318.

Michel Simard, Cyril Goutte, and Pierre Isabelle. 2007. Statistical Phrase-Based Post-Editing. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL HLT)*, pages 508–515, Rochester, New York.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A Study of Translation Edit Rate with Targeted Human Annotation. In *Proceedings of Association for Machine Translation in the Americas*, pages 223–231, Cambridge, Massachusetts, USA.

Antonio Toral. 2020. Reassessing Claims of Human Parity and Super-Human Performance in Machine Translation at WMT 2019. *arXiv preprint arXiv:2005.05738*.

Marco Turchi, Matteo Negri, and Marcello Federico. 2013. Coping with the subjectivity of human judgements in MT quality estimation. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 240–251, Sofia, Bulgaria. Association for Computational Linguistics.

Marco Turchi, Matteo Negri, and Marcello Federico. 2014. Data-driven annotation of binary MT quality estimation corpora based on human post-editions. *Machine Translation*, 28(3):281–308.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Jiayi Wang, Ke Wang, Kai Fan, Yuqi Zhang, Jun Lu, Xin Ge, Yangbin Shi, and Yu Zhao. 2020. Alibaba's Submission for the WMT 2020 APE Shared Task: Improving Automatic Post-Editing with Pre-trained Conditional Cross-Lingual BERT. In *Proceedings of the Fifth Conference on Machine Translation*, Online.

Hao Yang, Minghan Wang, Daimeng Wei, Hengchao Shang, Jiaxin Guo, Zongyao Li, Lizhi Lei, Ying Qin, Shimin Tao, Shiliang Sun, and Yimeng Chen. 2020. HW-TSC's Participation at WMT 2020 Automatic Post Editing Shared Task. In *Proceedings of the Fifth Conference on Machine Translation*, Online.

Barret Zoph and Kevin Knight. 2016. Multi-source neural translation. *arXiv preprint arXiv:1601.00710*.

Findings of the WMT 2020 Biomedical Translation Shared Task: Basque, Italian and Russian as New Additional Languages

Rachel Bawden[1*] **Giorgio Maria Di Nunzio**[2] **Cristian Grozea**[3]
Iñigo Jauregi Unanue[4] **Antonio Jimeno Yepes**[5] **Nancy Mah**[6]
David Martinez[5] **Aurélie Névéol**[7] **Mariana Neves**[8,9]
Maite Oronoz[10] **Olatz Perez de Viñaspre**[10] **Massimo Piccardi**[4]
Roland Roller[11] **Amy Siu**[12] **Philippe Thomas**[11] **Federica Vezzani**[13]
Maika Vicente Navarro[14] **Dina Wiemann**[15] **Lana Yeganova**[16]

[1]School of Informatics, University of Edinburgh, Scotland
[2]Dept. of Information Engineering, University of Padua, Italy
[3]Fraunhofer Institute FOKUS, Berlin, Germany
[4]University of Technology Sydney, Sydney, Australia
[5]IBM Research Australia, Melbourne, Australia
[6]Fraunhofer Institute for Biomedical Engineering (IBMT), Berlin, Germany
[7]LIMSI, CNRS, Université Paris-Saclay, Orsay, France
[8]German Centre for the Protection of Laboratory Animals (Bf3R),
[9]German Federal Institute for Risk Assessment (BfR), Berlin, Germany
[10]IXA NLP Group, University of the Basque Country, Donostia, Spain
[11]German Research Center for Artificial Intelligence (DFKI), Berlin, Germany
[12]Beuth University of Applied Sciences, Berlin, Germany
[13]Dept. of Linguistic and Literary Studies University of Padua, Italy
[14]Maika Spanish Translator, Melbourne, Australia
[15]Novartis AG, Basel, Switzerland
[16]NCBI/NLM/NIH, Bethesda, USA

Abstract

Machine translation of scientific abstracts and terminologies has the potential to support health professionals and biomedical researchers in some of their activities. In the fifth edition of the WMT Biomedical Task, we addressed a total of eight language pairs. Five language pairs were previously addressed in past editions of the shared task, namely, English/German, English/French, English/Spanish, English/Portuguese, and English/Chinese. Three additional languages pairs were also introduced this year: English/Russian, English/Italian, and English/Basque. The task addressed the evaluation of both scientific abstracts (all language pairs) and terminologies (English/Basque only). We received submissions from a total of 20 teams. For recurring language pairs, we observed an improvement in the translations in terms of automatic scores and qualitative evaluations, compared to previous years.

[*] The author list is alphabetical and does not reflect the respective author contributions.

1 Introduction

Automatic translation aims to alleviate the language barrier by providing access to information for readers not familiar with the original language used to write documents. Access to accurate biomedical information is specifically critical and machine translation (MT) can contribute to making health information available to health professionals and the general public in their own language. It can also contribute to biomedical research by assisting with the writing of research reports in English. In addition, machine translation can provide the opportunity to enhance the use of natural language processing (NLP) tools and methods for low-resource languages by the development of resources through translation or by making tools available through text translation into resource rich languages.

Herein, we describe the fifth edition of the WMT Biomedical task,[1] which aims to evaluate the auto-

[1] http://www.statmt.org/wmt20/

Proceedings of the 5th Conference on Machine Translation (WMT), pages 660–687
Online, November 19–20, 2020.

matic translation of a variety of biomedical texts.

The first edition of the task (Bojar et al., 2016) focused on biomedical scientific abstracts in three language pairs. The second edition of the task offered ten language pairs and addressed scientific abstracts as well as patient-oriented health information (Jimeno Yepes et al., 2017). The third edition of the task offered six language pairs and addressed scientific abstracts (Neves et al., 2018). The fourth edition of the task offered ten language pairs. It addressed scientific abstracts and introduced the task of terminology translation (Bawden et al., 2019). This year's edition of the task continues to address the translation of scientific abstracts and terminologies. It builds on previous tasks by offering a large range of training and test sets to support participants' systems. The following language pairs are addressed this year:

- English to Basque (en2eu)
- English to Chinese (en2zh) and Chinese to English (zh2en)
- English to French (en2fr) and French to English (fr2en)
- English to German (en2de) and German to English (en2de)
- English to Italian (en2it) and Italian to English (it2en)
- English to Portuguese (en2pt) and Portuguese to English (pt2en)
- English to Russian (en2ru) and Russian to English (ru2en)
- English to Spanish (en2es) and Spanish to English (es2en)

Similar to previous years, our test sets consist of scientific abstracts retrieved from the MEDLINE® database. In continuation with last year's task (Bawden et al., 2019), we also provide a test set for the automatic translation of biomedical terminologies. Below, we highlight some new aspects introduced in the 2020 edition of the shared task:

- We address three new language pairs, namely, en/eu, en/it, en/ru[2].
- We include a novel test set for the automatic translation of biomedical terminologies from English to Basque (cf. Section 2.2.1)
- During the construction of the test sets, and after the manual validation of the automatic alignment, we ran a pilot project for a couple of languages in which we manually fine-tuned the alignment of the test sets (cf. Section 2.2.3).
- We ran a second pilot study in which we split the sentences according to the reported original language of the abstract (cf. 2.2.3).
- Three of our tests sets, namely, de/en, ru/en and zh/en, were included as test suites in the WMT News Task (cf. Section 5.2).
- Participants were asked to provide details about their systems through an online survey (cf. Tables 6, 7, and 9).
- Our manual validation included whole abstracts, in addition to (correctly aligned) sentence pairs (cf. Section 6.1).
- We ran a third pilot study in which two experts validated submissions for certain language pairs, in which one was a native speaker of the source language, while the other a native speaker of the target language (cf. Tables 17 and Table 20).
- Our methodology for ranking the systems based on the manual validation considered a significance test and a points-based schema (cf. Section 6.1).

This article is structured as follows: Section 2 presents the details of the generation of our training and test sets, for both the scientific abstracts and the terminology, as well as manual validation of the quality of the test sets. Section 3 describes our baseline systems, which are used as comparison in the automatic evaluation. We list all teams that participated in our task in Section 4, as well as details of the methods behind their systems and the in-domain and out-of-domain data that was used. The results of the automatic evaluation based on the BLEU and chrF scores are presented in Section 5, while the ones for the manual evaluation are presented in Section 6. Finally, we discuss various topics related to the shared task in Section 7.

biomedical-translation-task.html

[2]Throughout the paper, we will refer to en/ru (or ru/en), for instance, when referring to the language pair in general, without specifying the translation direction. When making reference to the direction, we will use either en2ru or ru2en, for instance.

2 Training and test data

We provided training data of MEDLINE abstracts for it/en and ru/en, since training data for some of the other languages was already available from previous years. As for the tests sets, we released test sets for scientific abstracts and for terminologies, as summarized below:

- Scientific abstracts:
  - English to Basque
  - Chinese/English (both directions)
  - French/English (both directions)
  - German/English (both directions)
  - Italian/English (both directions)
  - Portuguese/English (both directions)
  - Spanish/English (both directions)
- Terms from biomedical terminologies:
  - English to Basque

Additional details are presented in Table 1. In this section we describe the details about the construction of resources that we released for the shared task.

2.1 Training data

We released training data from MEDLINE for two of the new language pairs that we address this year, namely, English/Italian and English/Russian.

We relied on the latest version of the MEDLINE baseline[3] available at the time of data preparation. We retrieved all the abstracts that were available in Italian and English, or in Russian and English. We summarize below the steps that we followed to process the data:

1. Abstracts were parsed using the `pubmed_parser` library.[4]
2. The language of these abstracts, as identified by MEDLINE meta-data, was confirmed with the `langdetect` library.[5]
3. Sentences in the abstracts were split using the `syntok` library.[6]
4. These sentences were automatically aligned using the `GMA` tool[7] using specific stopword lists for each language.

We obtained a total of 1,675 parallel documents for it/en and 6,029 for ru/en. The training data is available in our GitHub repository.[8]

In regard to English-Basque scientific abstract translation, we could not release any in-domain parallel data, as very little is still written in Basque in the medical domain. However, we provided other corpora that can help with training machine translation models. These include out-of-domain parallel corpora such as the TED talks,[9] the datasets available on the OPUS repository[10] and the WMT16 IT translation shared-task.[11] Additionally, we released in-domain monolingual corpora[12] that include translations of examples of hospital notes, automatic translations of SNOMED CT terms (Perez-de Viñaspre and Oronoz, 2015), and medical domain articles from Wikipedia. Finally, we released a recent dump of the whole Wikipedia (01/2020) as a large, out-of-domain monolingual corpus.[13]

For the terminology translation task, on behalf of Osakidetza (Basque Public Health System), we released 27,900 terms of the Basque ICD-10-CM. These descriptions were manually validated by the institution's translation team. 25,900 descriptions where released as a training set, keeping the remaining 2,000 for the development set. Both sets are plain text, and they have not been tokenized. On average, in the training set, each term comprises 6.72 words (split on whitespace and punctuation), 1 being the minimum and 27 the maximum. For the development set, the average word count is 6.75, 1 being the minimum and 25 the maximum.

2.2 Test sets

All test sets were released on June 29th, 2020 and the participants could submit results until July 9th, 2020. The test sets for de/en, ru/en and zh/en were

[3] https://www.nlm.nih.gov/databases/download/pubmed_MEDLINE.html released at the end of 2019.

[4] https://github.com/titipata/pubmed_parser

[5] https://pypi.org/project/langdetect/

[6] https://github.com/fnl/syntok

[7] https://nlp.cs.nyu.edu/GMA/

[8] https://github.com/biomedical-translation-corpora/corpora

[9] https://wit3.fbk.eu/mt.php?release=2018-01

[10] http://opus.nlpl.eu/

[11] http://www.statmt.org/wmt16/it-translation-task.html

[12] https://drive.google.com/drive/u/2/folders/1cQmiywDRcAeHeRuZfaF-zuoG7DQHO4CQ

[13] https://drive.google.com/drive/u/2/folders/1BjScNNvMbVOzrD3KWA0D0UGR33j6Lg83

Language pairs	MEDLINE training		Abstracts test		Terminology test
	Documents	Sentences	Documents	Sentences	Terms
en2eu	-	-	40	375	2,000
de2en	-	-	50	612/652	-
en2de			50	783/742	-
es2en	-	-	50	533/629	-
en2es			50	618/562	-
fr2en	-	-	50	563/584	-
en2fr			50	757/731	-
it2en	1,675	15,950/ (it)	50	549//716	-
en2it		20,615 (en)	50	624/468	-
pt2en	-	-	50	498/637	-
en2pt			50	544/466	-
ru2en	6,029	52,544/ (ru)	50	463/523	-
en2ru		61,494 (en)	50	553/484	-
zh2en	-	-	50	412/622	-
en2zh			50	514/343	-

Table 1: Number of documents, sentences and terms in the training and test sets released for this shared task.

also included as test suites of the WMT news task and released on June 22nd, 2020. In the following we describe details of the test set construction.

2.2.1 Terminology

In addition to the training set of ICD-10-CM Basque terms, there were 2,000 more terms for the test set. Again, this set was not tokenized. On average, each term comprises 7.74 words, 1 being the minimum word count and 25 the maximum. Unfortunately, at the time of releasing the test set, due to a confusion on behalf of the organizers, the development set was provided as test for all participants, and was used for evaluation. The planned test set has been publicly released for download.[14]

2.2.2 Basque abstracts

The Basque language appears in MEDLINE as a subject of study but not systematically as a writing language, so there is not a sufficient corpus for training in Basque in MEDLINE. The abstracts used in the test are taken from the journal Osagaiz,[15] the first journal on medicine written entirely in Basque (with abstracts also in English).

Osagaiz[16] was published for the first time in 2017 and every year it publishes a volume with at least two numbers. Its main objective is to be a way of communicating the scientific findings of the Basque health community in Basque. Three volumes have been used in the test (years 2017, 2018 and 2019); that is, 6 numbers with 40 abstracts in both English and Basque. The Basque abstracts dataset consists of 375 sentences (8,651 tokens in English with 23.07 tokens per sentence, and 7459 tokens in Basque with 19.89 tokens per sentence).

2.2.3 MEDLINE abstracts

We followed a similar approach to the one we used in previous years. However, we carried out two novel pilot studies this year: (a) a manual improvement of the alignment after the manual validation, and (b) a selective split of the abstracts for the translation directions based on the original language of the abstract.

For the test sets, we retrieved the citations that were published in 2020 and were not included in any of the previously released training and test sets. We parsed the articles and checked the language using the same tools as described for the training data above. We split the sentences for all languages using the `syntok` library, except for zh/en where it was sufficient to split sentences according to the Chinese punctuation (。) that marks the end of a sentence. Sentence alignment was carried out for all languages (except for zh/en) with the `GMA` tool using specific stopword lists for each language. For zh/en, we used the `Champollion` tool[17] with the same configurations and stopword lists since 2018.

[14] https://drive.google.com/drive/folders/1KXUjEBUzudi81y5rxm33UxkmRY9RSKMj

[15] http://www.osagaiz.eus/

[16] The contents from Osagaiz are licensed under Creative Common Attribution-ShareAlike 3.0 unported (CC BY-SA 3.0) https://creativecommons.org/licenses/by-sa/3.0/deed.en

[17] http://champollion.sourceforge.net/

We randomly retrieved a set of 100 abstracts for each language pair, and the automatic aligned sentences were manually validated by native speakers of the foreign languages using the Appraise tool (Federmann, 2010). Results of the validation are shown in Table 2. For the ru/en set, an additional set of 100 abstracts were randomly retrieved for a second round of manual validation. This was due to the low quality of the alignments that we obtained in the first round of validation. The official test set for ru/en was composed of the abstracts with better quality from the totality of 200 abstracts that were validated.

As a pilot study this year, we performed a manual correction of the alignment which were identified as not being correct during the validation in the Appraise tool. This step was only carried our for the es/en, fr/en, ru/en, and zh/en test sets. For all these languages, this extra step increased alignment quality (cf. Table 2): from 80.54% correctly aligned sentences to 91.49% for fr/en, from 55.27% to 61.96% for ru/en, from 83.57% to 88.07% for es/en, and a slight improvement from 63.84% to 64.43% for zh/en.

Most of the remaining sentences are in fact titles in English, for which a translation in the foreign language is not available from MEDLINE. For zh/en, the manual corrections addressed mismatching sentence splitting policies for abstract subsections such as *OBJECTIVE: To investigate...* and *METHODS: We used xyz...* The GMA tool split such a text into two sentences, but the Champollion tool kept it as one sentence. With this extra step, affected sentences that were marked as "NO_ALIGNMENT" became "TARGET_GREATER_SOURCE" (cf. Table 2 for the alignment categories).

Finally, the set of 100 abstracts was randomly split into two sets of 50 abstracts, for each translation direction, e.g., es2en and en2es. Exception was made for the fr/en test set. Following the recommendations of Graham et al. (2019), we tried to split the data sets depending on which language we hypothesized was the abstract's source language. For articles with a documented "TT" field (vernacular, i.e. French, title) in the MEDLINE citation, we considered that the source language was French and otherwise, English. As a result, the en/fr test only contains abstract originally written in English. However, since only 20 abstract in our set were originally written in French, the fr/en set still contains a mix of source languages. This suggests that vernacular titles should be considered in the initial set selection.

3 Baselines

We provided our baseline systems for all language pairs in the scientific abstracts translation subtask.

There were two categories of baseline: for en/zh, en/fr, en/de, en/pt and en/es the models used for each direction were transformers (Vaswani et al., 2017) trained by us using MarianNMT (Junczys-Dowmunt et al., 2018) with the following settings: joint BPE of 40,000, beam size 16. These parameters were chosen by tuning on a single direction of a single language pair: English to German. Each of the 10 models were trained for up to two days. The training was stopped when there were no improvements on the validation dataset for more than 10 epochs, as measured through cross-validation score. The corpora we used to train the models were the same as last year – when we had baselines generated using RNN-based sequence2sequence models: the UFAL medical corpus (UFA) without the "Subtitles" subset, and as validation we again used Khreshmoi (Dušek et al., 2017).

For en/it and en/ru and en/eu we used the Helsinki-NLP/opus-mt-SRC-TRG models (Tiedemann and Thottingal, 2020) included in the huggingface transformers library [18], trained with MarianNMT on the entirety of the OPUS corpora (Tiedemann, 2012). These models are not uniformly good; they performed very well for Italian, but fairly poor for Russian and Basque.

Discussion. It is interesting that the models for English to/from Italian performed so well in the biomedical task, as they were trained on generic text, not targeting the biomedical domain. It is interesting in general to what extent models that excel on generic text (e.g. news) perform well on the biomedical texts as well.

4 Teams and systems

This year, 22 teams submitted a total of 151 runs. Two teams withdrew after submitting their runs. The remaining teams were from China (7 teams), Spain (3 teams), France (2 teams), the United Kingdom (2 teams), Armenia (1 team), Australia (1 team), Brazil (1 team), India (1 team), Ireland (1 team) and Pakistan (1 team). Table 3 presents the

[18] https://huggingface.co/transformers/

Language	OK	Source>Target	Target>Source	Overlap	No Align.	Total
de/en	909 (70.85%)	63 (4.91%)	104 (8.11%)	52 (4.05%)	155 (12.08%)	1,283
es/en	931 (83.57%)	29 (2.60%)	54 (4.85%)	7 (0.63%)	93 (8.35%)	1,114
es/en §	1,026 (88.07%)	9 (0.78%)	4 (0.34%)	0 (0%)	126 (10.82%)	1,165
fr/en	985 (80.54%)	34 (2.78%)	74 (6.05%)	6 (0.49%)	124 (10.14%)	1,223
fr/en §	1225 (91.49%)	7 (0.52%)	8 (0.60%)	2 (0.15%)	97 (7.24%)	1,339
it/en	636 (60.40%)	51 (4.84%)	150 (14.25%)	60 (5.70%)	156 (14.81%)	1,053
pt/en	799 (78.41%)	37 (3.63%)	66 (6.48%)	20 (1.96%)	97 (9.52%)	1,019
ru/en *	947 (53.14%)	67 (3.76%)	186 (10.44%)	65 (3.65%)	517 (29.01%)	1,782
ru/en **	472 (55.27%)	33 (3.86%)	94 (11.01%)	32 (3.75%)	223 (26.11%)	854
ru/en §	562 (61.96%)	30 (3.3%)	60 (6.61%)	28 (3.09%)	228 (25.14%)	908
zh/en	535 (63.84%)	36 (4.30%)	135 (16.11%)	9 (1.07%)	123 (14.68%)	838
zh/en §	540 (64.43%)	137 (16.35%)	142 (16.95%)	9 (1.07%)	10 (1.19%)	838

Table 2: Statistics (number of sentences and percentages) of the quality of the automatic alignment for the MEDLINE test sets. For each language pair, the total number of sentences corresponds to the 100 documents that constitute the two test sets (one for each language direction). * Results for the totality (200 abstracts) for ru/en. ** Results for the selected test set (100 abstracts) for ru/en. § Results after manual correction of sentence segmentation and/or alignment.

list of teams that submitted at least one run to the biomedical task.

At least one run was submitted for each language pair offered, with the most runs submitted for English to Basque (terminology test set, 24 runs) and English to Chinese (MEDLINE test set, 18 runs). Table 4 presents an overview of the runs submitted by each team for language directions translating *from* English. Table 5 presents an overview of the runs submitted by each team for language directions translating *into* English.

During the automatic evaluation, we observed that some teams obtained extremely high BLEU scores, which were close to 0.9. Those teams had trained their systems on the MEDLINE database, and the training data potentially included our test sets. Unfortunately, as opposed to previous years, we forgot to inform participants on our website that this practice was not allowed. Therefore, we offered the opportunity for these teams to re-submit their runs, but without training on MEDLINE. The Wei-Bot team was the only one to submit new runs.

In an effort to increase the level of detail in the system description and the comparability between systems, we asked participants to fill in a survey with key information regarding the translation method used, as well as the in-domain and general datasets used for training. The survey comprised 14 questions covering the translation methods and corpora used. Teams indicated their primary submission, which was considered for manual evaluation. On average, submission time for one language pair was 6 minutes and 28 seconds (Median: 3 minutes and 35 seconds). All teams used transformer-based neural machine translation (except for team TRAMECAT, who used sequence2sequence) and mostly relied on existing implementations: 19 teams submitted runs using available libraries, one team submitted runs using a mix of libraries and in-house implementations, one team submitted runs exclusively relying on their own implementation of NMT. Teams often used the same setup for a range of language pairs. Table 6 shows details about the teams methods.

For in-domain data, teams used the training data distributed by us and many of the sources described in (Névéol et al., 2018). Tables 7 and 8 provide details of the in-domain data used by the teams.

For relevant language pairs, parallel data from other WMT tracks (e.g., News Task) was used. Interestingly, some teams used similarity measures based on biomedical corpora to extract additional biomedical sentences from out-of-domain corpora. Out-of-domain data was also used in the form of pre-trained base models. Table 9 shows details of the out-of-domain data used by the teams.

5 Automatic evaluation

Following (Mathur et al., 2020), we used chrF (Popović, 2015) as well as BLEU (Papineni et al., 2002) as automatic metrics. chrF scores are obtained using the `nltk` implementation.[19]

5.1 MEDLINE

Similarly to previous years, we compared the submitted translations to the reference translations

[19] https://www.nltk.org/_modules/nltk/translate/chrf_score.html

Team ID	Institution
ADAPT (Nayak et al., 2020)	Dublin City University, Ireland
ai_not_intellegent	ai_not_intellegent, China
Alibuba	Alibab DAMO Academy, China
baidu_translation	Baidu translation, China
Elhuyar_NLP (Corral and Saralegi, 2020)	Elhuyar Foundation, Spain
Huawei United (Peng et al., 2020)	Huawei Technologies, China
Ixamed (Soto et al., 2020)	University of the Basque Country, Spain
LIMSI (Abdul Rauf et al., 2020)	LIMSI-CNRS, France
NLE	Naver Labs Europe, France
nrpu-fjwu (Naz et al., 2020)	Fatima Jinnah Women University, Pakistan
one_connect_000	OneConnect AI Lab, China
OOM_20	Atman Tech, India
Sheffield (Soares and Vaz, 2020)	University of Sheffield, UK
TMT (Wang et al., 2020)	Tencent AI Lab, China
TRAMECAT	Universitat Oberta de Catalunya, Spain
UNICAM (Saunders and Byrne, 2020)	University of Cambridge, UK
UNICAMP_DL (Lopes et al., 2020)	University of Campinas, Brazil
UTS_NLP (Jauregi Unanue and Piccardi, 2020)	University of Technology Sydney, Australia
Wei-Bot	East China Normal University, China
YerevaNN (Hambardzumyan et al., 2020)	YerevaNN, Armenia

Table 3: List of the participating teams.

Teams	en2eu	en2de	en2es	en2fr	en2it	en2pt	en2ru	en2zh	Total
ADAPT	A3T3	-	-	-	-	-	-	-	6
ai_not_intellegent	-	-	-	-	-	-	-	A3	3
Alibuba	-	-	-	-	-	-	-	A1	1
baidu_translation	-	-	-	-	-	-	-	A1	1
Elhuyar_NLP	A3T3	-	A3	-	-	-	-	-	9
Huawei United	-	A3	-	A2	A2	-	A2	A3	12
Ixamed	A3T3	-	A3	-	-	-	-	-	9
LIMSI	-	-	-	A2	-	-	-	-	2
NLE	-	A3	-	-	-	-	-	-	3
nrpu-fjwu	-	-	-	A1	-	-	-	-	1
one_connect_000	-	-	-	-	-	-	-	A1	1
OOM	-	-	-	-	-	-	-	A2	2
Sheffield	-	-	A1	A1	A1	A1	A1	-	5
TMT	-	A3	-	-	-	-	-	A3	6
TRAMECAT	-	-	A1	A1	-	-	A1	A1	4
UNICAM	-	A3	A3	-	-	-	-	-	6
UNICAMP	-	-	-	-	-	A2	-	-	2
UTS_NLP	A3T3	-	-	-	-	-	-	-	6
Wei-Bot	-	-	-	-	-	-	-	A2	2
YerevaNN	-	A2	-	-	-	-	A3	-	5
Total	24	14	11	7	3	3	7	17	86

Table 4: Overview of the submissions from all teams and test sets translating from English. We identify submissions to the abstracts testsets with an "A" and to the terminology test set with a "T". The value next to the letter indicates the number of runs for the corresponding test set, language pair, and team.

using BLEU with the `MULTI-EVAL v14` tool[20] provided by the Moses package (Koehn et al., 2007). This means as well that we reused the tokenization approach used for Chinese. Results for MEDLINE BLEU are shown in Tables 10 and 11.

[20] https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/mteval-v14.pl

5.2 News

The test set of our challenge was included in the News challenge data set. We identified the translations in the News files and used the same evaluation procedure as applied to MEDLINE abstracts. Results of the systems are shown in Tables 12 and 13.

Teams	de2en	es2en	fr2en	it2en	pt2en	ru2en	zh2en	Total
ai_not_intellegent	-	-	-	-	-	-	A3	3
Alibuba	-	-	-	-	-	-	A1	1
baidu_translation	-	-	-	-	-	-	A1	1
Huawei United	A3	-	A2	A2	-	A2	A2	11
Ixamed	-	A3	-	-	-	-	-	3
NLE	A3	A1	A1	A1	-	-	-	6
nrpu-fjwu	-	-	A3	-	-	-	-	3
one_connect_000	-	-	-	-	-	-	A1	1
OOM	-	-	-	-	-	-	A2	2
Sheffield	-	A1	A1	A1	A1	A1	-	5
TMT	A3	-	-	-	-	-	A1	4
TRAMECAT	-	A1	A1	-	-	A1	A1	4
UNICAM	A3	A3	-	-	-	-	-	6
UNICAMP	-	-	-	-	A2	-	-	2
Wei-Bot	-	-	-	-	-	-	A2	2
YerevaNN	A3	-	-	-	-	A2	-	5
Total	15	9	8	4	3	6	14	59

Table 5: Overview of the submissions from all teams and test sets translating into English. We identify submissions to the abstracts test sets with an "A" and to the terminology test set with a "T". The value next to the letter indicates the number of runs for the corresponding test set, language pair, and team.

Team ID	Language pair	NMT implementation	Trained	Fine-Tuned	BT	LM
ADAPT	all	Marian NMT	Yes	No	Yes	No
ai_not_intellegent	zh2en	Fairseq	Yes	Yes	No	No
ai_not_intellegent	en2zh	Own	No	Yes	No	MASS
Alibuba	zh2en	OpenNMT	Yes	No	Yes	transformer-base
Alibuba	en2zh	OpenNMT	No	Yes	Yes	transformer-base
baidu_translation	all	paddle	Yes	No	Yes	paddle
Elhuyar_NLP	all	OpenNMT	Yes	No	en2eu	No
Huawei United	en/de	Own	Yes	No	No	FB-PLM
Huawei United	all but en/de	Own	Yes	No	zh2en	No
Ixamed	all	Open NMT	Yes	No	No	No
LIMSI	all	Fairseq	Yes	Yes	Yes	Yes
NLE	de2en	Fairseq	Yes	No	Yes	No
NLE	fr2en	Fairseq	Yes	Yes	Yes	No
NLE	it2en	Fairseq	Yes	Yes	Yes	No
nrpu-fjwu	all	Fairseq	Yes	No	Yes	fr2en
OOM_20	all	tensor2tensor, modified	Yes	Yes	-	-
Sheffield	all but ru/en	Tensorflow	Yes	No	{es,fr,it,pt}2en	No
Sheffield	ru2en, en2ru	Tensorflow	Yes	Yes	ru2en	No
TMT	all	Fairseq	Yes	No	Yes	No
TRAMECAT	all	MarianNMT	Yes	No	No	No
UNICAM	all	Tensor2Tensor	No	Yes	No	No
UNICAMP_DL	all	T5, Huggingface	No	Yes	No	T5 HuggingFace
UTS_NLP	all	Fairseq, BERT-NMT	Yes	No	Yes	Yes
Wei-Bot	all	Fairseq	Yes	No	Yes	MASS
YerevaNN	all	Fairseq?	No	Yes	ru2en	XLM-R

Table 6: Overview of methods used by participating teams. Information is self-reported through our survey for each selected "best run". BT indicates if backtranslation is used and LM if language models were used.

5.3 Basque abstracts

For the Basque abstract we used the same evaluation tool as for MEDLINE (MULTI-EVAL), and the results are presented in Table 14.

5.4 Terminology

For the evaluation of terminology we provide two metrics for the en2eu task: (i) accuracy, by relying on strict matches (case-insensitive) between ground truth and predictions; and (ii) sentence-level BLEU score, as measured by the `nltk` module sentence-BLEU.[21] Results are presented in Table 15.

[21] https://www.nltk.org/_modules/nltk/translate/bleu_score.html

Language pair	team	Parallel corpus	size (sentence pairs)	Monolingual corpus	size (sentences)
en/de	Huawei	MEDLINE abstracts corpus supplied by organizers.	29 k	No	-
	NLE	MEDLINE abstracts corpus supplied by organizers.	34,710	No	-
	UNICAM	TRAINING: UFAL medical and MEDLINE abstracts corpus supplied by organizers. FINE-TUNING: MEDLINE abstracts	TRAINING: 2.2M FINE-TUNING: 28K	No	-
	TMT	UFAL medical and MEDLINE abstracts corpus supplied by organizers.	2.5M	UFAL (en)	5.4M
	Yereva_NN	MEDLINE abstracts corpus supplied by organizers; alignment was fixed using XLM-R	32,466	No	-
en/es	Elhuyar_NLP	Scielo and corpora supplied by organizers.	560k	No	-
	Ixamed	MEDLINE corpus supplied by organizers and TAUS Corona Crisis Corpus	1,290,201	No	-
	UNICAM	TRAINING: UFAL medical, Scielo (Neves et al., 2016), and MEDLINE abstracts corpus supplied by organizers. FINE-TUNING: MEDLINE abstracts	TRAINING: 1.3M FINE-TUNING: 67K	No	-
	Sheffield	BVS, EMEA, Scielo (Soares et al., 2018) and MEDLINE corpus supplied by organizers as well as new crawled PubMed data. The data was checked against the official test set to avoid including test data during training.	2.5M	No	-
	TRAMECAT	Biomedical translation repository, EMEA, IBECS, ICD10, Kreshmoi, MEDLINE corpus supplied by organizers, in-house MEDLINE (dated 2018), Medem glossaries, MSDManuals, Portal Clinic corpus, Scielo, SNOMED	7,232,784	No	-
en/eu	ADAPT	Data provided by the organisers	-	Common Crawl selected by TermFinder	200k (en) 41,151 (eu)
	Elhuyar_NLP	WMT20 shared task bilingual training data, internal medical corpus, and synthetically generated data from the WMT19 EN-ES shared task	Around 350k segments	SNOMED descriptions, hospital notes and wikipedia medical articles (en)	Around 110k segments
	Ixamed	-	-	-	-
	UTS_NLP	ICD-10 codes translations	25900	SNOMED terms, hospital notes and wikipedia medical articles (en)	total of 60,000 sentences)
en/fr	Huawei	MEDLINE abstracts corpus supplied by organizers, in-domain lexicon	4M bitext, 59k lexicon	Yes (en)	22M
	LIMSI	Cochrane, Taus and corpora supplied by organizers	3,951,013	LISSA (Griffon et al., 2017) (fr)	395,699
	NLE	In-domain parallel data obtained from WMT and OPUS	-	No	-
	nrpu-fjwu	Corpora supplied by organizers (MEDLINE, Scielo, EDP, UFAL).	3,408,327	No	-
	Sheffield	EMEA and MEDLINE corpus supplied by organizers as well as new crawled PubMed data. Prior to training, the data was checked against the official test set to avoid including test data during training.	3.42M	MEDLINE (en)	2M
	TRAMECAT	EMEA, MEDLINE corpus supplied by organizers, PatTR medical, Scielo(Neves et al., 2016)	4,2 M	No	-
en/it	Huawei	MEDLINE abstracts corpus supplied by organizers.	219k	No	-
	NLE	MEDLINE corpus supplied by organizers, TAUS Corona Corpus, OPUS	-	No	-
	Sheffield	EMEA and MEDLINE corpus supplied by organizers as well as new crawled PubMed data. The data was checked against the official test set to avoid including test data during training.	1.0M	MEDLINE (en)	1M
en/pt	Sheffield	BVS, EMEA, Scielo (Soares et al., 2018) and MEDLINE corpus supplied by organizers as well as new crawled PubMed data. The data was checked against the official test set to avoid including test data during training.	5.5M	MEDLINE (en)	2M
	UNICAMP_DL	EMEA corpus, MEDLINE corpus supplied by organizers, Scielo (Soares et al., 2018), a corpus of theses and dissertations abstracts (BDTD) from CAPES, JRCAcquis.	6,606,858	MEDLINE (en)	2M

Table 7: Overview of in-domain corpora used by participating teams. Information is self reported through our survey for each selected "best run".

Language pair	team	Parallel corpus	size (sentence pairs)	Monolingual corpus	size (sentences)
en/ru	Huawei	MEDLINE abstracts corpus supplied by organizers.	32 k	No	-
	Sheffield	MEDLINE corpus supplied by organizers as well as new crawled PubMed data. The data was checked against the official test set to avoid including test data during training.	15k	MEDLINE (en)	100k
	TRAMECAT	MEDLINE corpus supplied by organizers, Corona TAUS corpus, ICD10 (subset)	240,998	No	-
	Yereva_NN	MEDLINE abstracts corpus supplied by organizers; alignment was fixed using XLM-R	37,201	No	-
en/zh	ai_not_intel...	Web crawl augmented by back translation	"3.G in text"	Yes	
	Alibuba	PubMed articles in Chinese	"1.2G text"	No	-
	Baidu	"inhouse dataset"	"12.5G text"	No	-
	Huawei	in-domain lexicon	59k	Yes (en)	62M
	OOM_20	Abstracts from Chinese medical papers	3 M	medical papers	(zh) 10M, (en) 20M
	TMT	No	-	Yes (en)	5.4M
	TRAMECAT	Corona TAUS corpus	450,507	No	-
	Wei-Bot	Pubmed Crawl	3M	Wikipedia (en, zh)	-

Table 8: (Continued...) Overview of in-domain corpora used by participating teams. Information is self reported through our survey for each selected "best run".

6 Manual evaluation

We manually validated a sample for each primary run in order to compare the performance between teams as well as to the reference translations. In this section we present details of the evaluation and results that we obtained.

6.1 MEDLINE abstracts

Similarly to previous years, we aimed to validate a total of 100 sampled sentences per primary run. This year, we manually validated not only single sentences, but also whole abstracts. The selection of abstracts to be validated for each language pair followed the procedure described below:

1. Randomly select an abstract.
2. Check whether the percentage of perfectly aligned sentences is at least of 80%.
3. Retrieve all perfectly (i.e., OK) aligned sentences from the abstract.
4. Repeat steps 1 to 3 above if the total number of selected sentences (over all selected abstracts) is below 100.

In the case of zh2en and en2zh, due to the large number of submissions that we received, the manual validation was restricted to the abstracts. However, these were selected using the same approach described above. In addition, one team re-submitted their results after the official test period, and we note that these re-submissions are not fully comparable to the ones submitted before the period (see Tables 10, 11 and 22).

Due to time constraints, we could not validate all planned abstracts and sentences that were selected for de2en, but only about half of them. Further, and due to the same reason, the validation for es2en and pt2en was limited to a few abstracts (and its sentences) and was validated as a collaboration between two experts: (1) one who was a native speaker of the source language and who checked whether any information that was included in the source text was missing in the translation; and (2) one who was a native speaker of English, and who was in charge of checking the quality of the English translations.

If the information about the primary run was not available for a particular team and test set, we considered the run with the highest BLEU score. We only considered for manual validation those teams that provided detailed information about their system by filling out a survey mentioned in Section 4. The runs that we considered are listed below:

- en2de (5 teams): Huawei United (run3), NLE (run3), TMT (run1), UNICAM (run3), YerevaNN (run3)
- en2es (5 teams): Elhuyar (run1), Ixamed (run1), Sheffield (run1), TRAMECAT (run1), UNICAM (run3)
- en2fr (5 teams): Huawei United (run2), LIMSI (run1), Sheffield (run1), TRAMECAT (run1), nrpu-fjwu (run1)

Language pair	team	Parallel corpus	size (sentence pairs)	Monolingual corpus	size (sentences)
en/de	Huawei	TRAINING: in-house bitext FINE-TUNING: tfidf filtering of training corpus	TRAINING: 2.3M FINE-TUNING: 27K	Yes (de)	2.3M
	NLE	All de-en parallel data supplied by WMT20 News Task	44.8M	NewsCrawl	269M (en) 440M (de)
	UNICAM	For pre-training, corpus supplied by the WMT 2018 news task organizers	17M	No	-
	TMT	Corpus supplied by the WMT 2020 News task organizers	37.8M	No	-
	Yereva_NN	No OOD data was used directly, but the base models we had fine-tuned were trained on news data (Ng et al., 2019)	-	No	-
en/es	Elhuyar_NLP	Paracrawl v5 corpus	33M	No	-
	Ixamed	No	-	No	-
	UNICAM	No	-	No	-
	Sheffield	No	-	No	-
	TRAMECAT	UNPC parallel corpus: segments selected by similarity (using a language model on the English part)	5M	No	-
en/eu	ADAPT	Data provided by the organisers	-	CommonCrawl (eu)	400K
	Elhuyar_NLP	Synthetic data was obtained by backtranslating an internal ES-EU corpus from Spanish to English	Around 7M segments	No	-
	Ixamed	-	-	-	-
	UTS_NLP	Out of domain parallel corpora provided by WMT2020 biomedical translation organizers.	approx. 0.6M	Wikipedia (eu)	1.5M
en/fr	Huawei	news and other data (in-house)	123M	Yes (en)	62M
	LIMSI	No	-	No	-
	NLE	OOD WMT and OPUS	-	Back Translation en2ko	8M
	nrpu-fjwu	Medical domain sentences retrieved from books, news commentary and wikiPedia parallel corpus.	243,182	medical sentences retrieved from wikiPedia (fr)	-
	Sheffield	No	-	No	-
	TRAMECAT	UNPC parallel corpus: segments selected by similarity (using a language model on the English part)	5M	No	-
en/it	Huawei	in-house general domain data like news	150M	No	-
	NLE	Paracrawl, OPUS, UN Political corpus	-	English sentences back-translated	9.2M
	Sheffield	No	-	No	-
en/pt	Sheffield	No	-	No	-
	UNICAMP_DL	ParaCrawl dataset (subset)	5M	No	-
en/ru	Huawei	No	-	No	-
	Sheffield	ParaPat corpus of Patents (Soares et al., 2020)	4.3M	MEDLINE (en)	100k
	TRAMECAT	UNPC parallel corpus: segments selected by similarity (using a language model on the English part)	5M	No	-
	Yereva_NN	No OOD data was used directly, but the base models we had fine-tuned were trained on news data (Ng et al., 2019)	-	No	-
en/zh	ai_not_intel...	Corpus supplied by the WMT 2020 News task organizers	"3.G in text"	No	-
	Alibuba	No	-	No	-
	Baidu	No	-	No	-
	Huawei	"inhouse dataset"	186M	No	-
	OOM_20	Corpus supplied by the WMT 2020 News task organizers	10 M	No	-
	TMT	No	-	Yes (en)	5.4M
	TRAMECAT	UNPC parallel corpus: segments selected by similarity (using a language model on the English part)	5M	No	-
	Wei-Bot	No	-	No	-

Table 9: Overview of out-of-domain (OOD) corpora used by participating teams. Information is self reported through our survey for each selected "best run".

- en2it (2 teams): Huawei United (run2), Sheffield (run1)
- en2pt (2 teams): Sheffield (run1), UNICAMP_DL (run1)
- en2ru (4 teams): Huawei United (run2), Sheffield (run1), TRAMECAT (run1), YerevaNN (run3)
- en2zh (8 teams): ai_not_intellegent (run1),

Teams	Runs	en2de	en2es	en2fr	en2it	en2pt	en2ru	en2zh
Alibuba	Run1	-	-	-	-	-	-	0.3346*
Elhuyar_NLP	Run1	-	0.4498*	-	-	-	-	-
	Run2	-	0.4493	-	-	-	-	-
	Run3	-	0.4394	-	-	-	-	-
Huawei_United	Run1	0.3317	-	0.4351*	0.4257	-	0.3464	0.4378
	Run2	0.362	-	0.4351*	0.4257*	-	0.3464*	0.4546
	Run3	0.3689*	-	-	-	-	-	0.4378*
Ixamed	Run1	-	0.4171*	-	-	-	-	-
	Run2	-	0.3836	-	-	-	-	-
	Run3	-	0.3858	-	-	-	-	-
LIMSI	Run1	-	-	0.3837*	-	-	-	-
	Run2	-	-	0.3673	-	-	-	-
	Run3	-	-	0.2564	-	-	-	-
NLE	Run1	0.3641	-	-	-	-	-	-
	Run3	0.3394	-	-	-	-	-	-
	Run3	0.3562*	-	-	-	-	-	-
OOM_20	Run1	-	-	-	-	-	-	0.4686*
	Run2	-	-	-	-	-	-	0.4633*
Sheffield	Run1	-	0.4493*	0.3049*	0.2073*	0.4744*	0.2573*	-
TMT	Run1	0.3524*	-	-	-	-	-	0.3943*
	Run2	0.3495	-	-	-	-	-	-
	Run3	0.3457	-	-	-	-	-	-
TRAMECAT	Run1	-	0.4361*	0.3489*	-	-	0.2661*	0.2725*
UNICAMP_DL	Run1	-	-	-	-	0.4095*	-	-
	Run2	-	-	-	-	0.3660	-	-
UNICAM	Run1	0.3288	0.4572	-	-	-	-	-
	Run2	0.3282	0.4672	-	-	-	-	-
	Run3	0.3318*	0.4662*	-	-	-	-	-
Wei-Bot	Run1	-	-	-	-	-	-	0.5557*§
	Run2	-	-	-	-	-	-	0.5169§
YerevaNN	Run1	0.3517	-	-	-	-	0.3263	-
	Run2	-	-	-	-	-	0.3936	-
	Run3	0.3520*	-	-	-	-	0.3787*	-
ai_not_intellegent	Run1	-	-	-	-	-	-	0.4462
	Run2	-	-	-	-	-	-	0.4148
	Run3	-	-	-	-	-	-	0.4225
baidu_translation	Run1	-	-	-	-	-	-	0.3400
nrpu-fjwu	Run1	-	-	0.3572*	-	-	-	-
one_connect_000	Run1	-	-	-	-	-	-	0.3125*
Baseline	-	0.2845	0.3813	0.3345	0.3954	0.4149	0.2259	0.2319

Table 10: BLEU scores for “OK" aligned test sentences, from English. * Indicates the primary run as indicated by the participants. § Runs submitted after the official test period.

Alibuba (run1), baidu_translation (run1), Huawei United (run3), OOM_20 (run1), TMT (run1), TRAMECAT (run1), Wei-Bot (run1)

- de2en (5 teams): Huawei United (run3), NLE (run3), TMT (run1), UNICAM (run3), YerevaNN (run3)
- es2en (4 teams): Ixamed (run1), Sheffield (run1), TRAMECAT (run1), UNICAM (run3)
- fr2en (5 teams): Huawei United (run2), NLE (run1), Sheffield 8run1), TRAMECAT (run1), nrpu-fjwu (run1)
- it2en (3 teams): Huawei United (run2), Sheffield (run1), NLE (run1)
- pt2en (2 teams): Sheffield (run1), UNICAMP_DL (run1)
- ru2en (4 teams): Huawei United (run2), Sheffield (run1), TRAMECAT (run1), YerevaNN (run3)
- zh2en (8 teams): ai_not_intellegent (run1), Alibuba (run1), baidu_translation (run1), Huawei United (run3), OOM_20 (run1), TMT (run1), TRAMECAT (run1), Wei-Bot (run1)

In addition to the above teams, we also considered the reference translation in the manual validation. We refer to these translations as validation *items* from here on. The selected sentences and abstracts were uploaded into the Appraise tool (Federmann, 2010) for manual validation. The valida-

Teams	Runs	de2en	es2en	fr2en	it2en	pt2en	ru2en	zh2en
Alibuba	Run1	-	-	-	-	-	-	0.2425*
Huawei_United	Run1	0.3897	-	0.4445	0.4974	-	0.4303	0.3378
	Run2	0.4146	-	0.4445*	0.4974*	-	0.4303*	0.3397
	Run3	0.4133	-	-	-	-	-	0.3528
Ixamed	Run1	-	0.4072*	-	-	-	-	-
	Run2	-	0.4073	-	-	-	-	-
	Run3	-	0.3999	-	-	-	-	-
NLE	Run1	0.4043	0.5075*	0.4349*	0.5011*	-	-	-
	Run2	0.4059	-	-	-	-	-	-
	Run3	0.4094*	-	-	-	-	-	-
OOM_20	Run1	-	-	-	-	-	-	0.3483*
	Run2	-	-	-	-	-	-	0.3473*
Sheffield	Run1	-	0.4624*	0.3514*	0.2276*	0.5334*	0.2936*	-
TMT	Run1	0.4165*	-	-	-	-	-	0.3048*
	Run2	0.4037	-	-	-	-	-	0.2893
	Run3	0.4080	-	-	-	-	-	0.2765
TRAMECAT	Run1	-	0.4468*	0.3477*	-	-	0.3707*	0.1688*
TXT	Run1	-	-	-	-	-	-	0.3048*
	Run2	-	-	-	-	-	-	0.2893
	Run3	-	-	-	-	-	-	0.2765
UNICAMP_DL	Run1	-	-	-	-	0.4988*	-	-
	Run2	-	-	-	-	0.4361	-	-
UNICAM	Run1	0.3962	0.4662	-	-	-	-	-
	Run2	0.3979	0.4640	-	-	-	-	-
	Run3	0.3963*	0.4657*	-	-	-	-	-
Wei-Bot	Run1	-	-	-	-	-	-	0.4009*§
	Run2	-	-	-	-	-	-	0.3946§
YerevaNN	Run1	0.4129	-	-	-	-	-	-
	Run2	0.4144	-	-	-	-	0.4331	-
	Run3	0.4128*	-	-	-	-	0.4321*	-
ai_not_intellegent	Run1	-	-	-	-	-	-	0.3357
	Run2	-	-	-	-	-	-	0.3226
	Run3	-	-	-	-	-	-	0.3323
baidu_translation	Run1	-	-	-	-	-	-	0.2494
nrpu-fjwu	Run1	-	-	0.2624*	-	-	-	-
	Run2	-	-	0.2273	-	-	-	-
	Run3	-	-	0.2041	-	-	-	-
one_connect_000	Run1	-	-	-	-	-	-	0.2238*
Baseline	-	0.3470	0.3534	0.3458	0.4588	0.4549	0.2984	0.1561

Table 11: BLEU scores for "OK" aligned test sentences, into English. * Indicates the primary run as indicated by the participants. § Runs submitted after the official test period.

	de2en	en2de	ru2en	en2ru	zh2en	en2zh
AFRL	-	0.2652	0.2895	-	-	-
ariel197197	-	-	0.2999	0.2270	-	-
DeepMind	-	-	-	-	0.2907	-
DiDi_NLP	-	-	-	-	-	-
eTranslation	-	0.257	0.3077	-	-	-
Huoshan_Translate	0.3287	0.2781	-	-	-	-
Online-A	0.3164	0.2649	0.2926	0.2115	0.2413	0.3431
Online-B	0.3342	0.2851	0.3514	0.2594	0.3041	0.3817
Online-G	0.3402	0.2536	0.335	0.2934	0.2854	0.3587
Online-Z	0.2786	0.2172	0.2379	0.1903	0.2162	0.2867
OPPO	0.3287	0.2792	0.3241	0.2566	0.3012	0.3908
PROMT_NMT	0.3100	0.2648	0.3230	0.2502	-	-
SJTU-NICT	-	-	-	-	0.3034	0.4159
Tencent_Translation	-	-	-	-	-	-
Tohoku-AIP-NTT	0.3411	0.2797	-	-	-	-
UEDIN	0.3160	0.2411	-	-	-	-
WMTBiomedBaseline	0.2865	0.2443	-	-	0.1529	-
yolo	0.0022	-	-	-	-	-
zlabs-nlp	0.2516	0.2225	0.2403	0.2016	0.2159	0.2868
Total	12	13	10	8	9	7

Table 12: BLEU scores for news test sentences

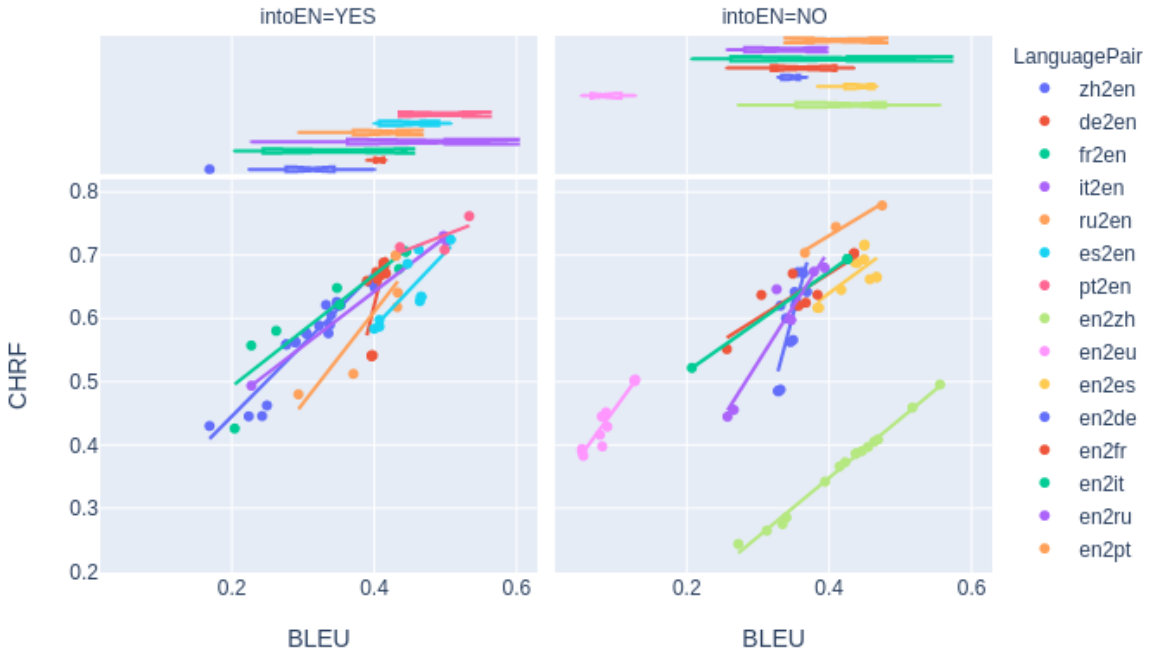


Figure 1: Fitted plot of BLEU vs. chrF scores for “OK" aligned test sentences, into English (left) and from English (right). The top section of the figure shows box plots of the BLEU score distribution for each language pair.

	de2en	en2de	ru2en	en2ru	zh2en	en2zh
AFRL	-	0.3193	0.3847	-	-	-
ariel197197	-	-	0.3911	0.3075	-	-
DeepMind	-	-	-	-	0.3015	-
DiDi_NLP	-	-	-	-	-	-
eTranslation	-	0.3097	0.4008	-	-	-
Huoshan_Translate	0.3915	0.3401		-	-	-
Online-A	0.3739	0.3229	0.3799	0.2926	0.2515	0.3723
Online-B	0.4009	0.3471	0.4711	0.3611	0.3210	0.4138
Online-G	0.3994	0.3086	0.4410	0.4089	0.2906	0.3897
Online-Z	0.3347	0.2546	0.3154	0.2587	0.2203	0.3096
OPPO	0.3915	0.3378	0.4239	0.3529	0.3166	0.4227
PROMT_NMT	0.3693	0.3167	0.4199	0.3434	-	-
SJTU-NICT	-	-	-	-	0.3217	0.4508
Tencent_Translation	-	-	-	-	-	-
Tohoku-AIP-NTT	0.4016	0.3388	-	-	-	-
UEDIN	0.3727	0.2922	-	-	-	-
WMTBiomedBaseline	0.3727	0.2864	-	-	0.1565	-
yolo	0.0026	-	-	-	-	-
zlabs-nlp	0.2961	0.2711	0.3188	0.2815	0.2277	0.3035
Total	12	13	10	8	9	7

Table 13: BLEU scores for news test “OK” sentences

tors were native speakers of the target language and had good knowledge of the source language. Each validator was presented with the source sentence (or abstract), and two candidate translations, either from two teams or from one team and the reference translation. The goal of the validator was to decide whether one translation was better than the other or whether they were of similar quality. Sentences could be skipped if the translations seemed to refer to different source sentences. Results for the manual validation are presented in various tables as summarized below:

- en2de and de2en: Table 16
- en2es and es2en: Table 17

Teams	Runs	BLEU
DCU-MT	Run1	0.0867
	Run2	0.0825
	Run3	0.0808*
Elhuyar_NLP	Run1	0.1271*
	Run2	0.1279
	Run3	0.1268
Ixamed	Run1	0.0815*
	Run2	0.0782
	Run3	0.0884
UTS_NLP	Run1	0.0530*
	Run2	0.0549
	Run3	0.0528
Baseline	-	0.0596

Table 14: Results for the abstract test set (en2eu). * indicates the primary run as indicated by the participants.

Teams	Runs	Accuracy	BLEU
Elhuyar_NLP	run1*	0.78	0.7373
	run2	0.77	0.7356
	run3	0.75	0.7229
ADAPT	run1	0.73	0.7083
	run2	0.76	0.7239
	run3	0.75	0.7179
UTS_NLP	run1*	0.73	0.7115
	run2	0.73	0.7122
	run3	0.73	0.7085
Ixamed	run1	0.12	0.1314
	run2*	0.08	0.0721
	run3	0.13	0.1481

Table 15: Results for the terminology test set (en2eu). * indicates the primary run as indicated by the participants.

- en2fr and fr2en: Table 18
- en2it and it2en: Table 19
- en2pt and pt2en: Table 20
- en2ru and ru2en: Table 21
- en2zh and zh2en: Table 22

We identified the item of each pairwise comparison (if any) that performed better (cf. respective tables) and ran a Wilcoxon Signed-Rank Test using the Python `scipy` library (Virtanen et al., 2020). We consider all comparisons for two particular items over all validated segments (abstracts and sentences), except for skipped segments. The test was calculated for the abstracts and the sentences and we mark in bold in the respective tables if any of them was found to be significant, (p-value< 0.05), otherwise, the two items were considered to be similar. For the language pairs validated by two experts (i.e., es2en and pt2en), we only consider one item of the pairwise comparison to be superior to the other when at least two of the four comparisons (2x for the abstracts, 2x for the sentences) were statistically significant.

To rank the systems, we assign points to each item: 3 points if superior to the opponent, 1 point when they are similar and no points if inferior to the opponent. Based on this methodology, we ranked the systems and the reference translations as summarized below (the obtained points are shown in parentheses):

- en2de: UNICAM (1) < reference (5) < YerevaNN (6) < Huawei-United (7) = NLE (7) < TMT (9)
- en2es: reference (2) < TRAMECAT (4) < Sheffield (5) < Ixamed (6) = UNICAM (6) < Elhuyar_NLP (11)
- en2fr: Sheffield (2) < TRAMECAT (3) = LIMSI (3) < nrpu-fjwu (5) < Huawei United (12) < reference (15)
- en2it: Sheffield (0) < reference (4) = Huawei United (4)
- en2pt: UNICAMP_DL (0) < reference (3) = Sheffield (3)
- en2ru: Sheffield (1) < TRAMECAT (2) < Huawei United (4) < YerevaNN (9) < reference (12)
- en2zh: TRAMECAT (1) < TMT (6) < baidu (10), ai_not_intellegent (10) < Wei-Bot (12) = OOM (12) = Huawei United (12) = Alibuba (12) = reference (12)
- de2en: UNICAM (2) < TMT (5) = reference (5) < Huawei United (7) = YerevaNN (7) = NLE (7)
- es2en: reference (5) = Ixamed (5) = NLE (5) = Sheffield (5) = TRAMECAT (5) = UNICAM (5)
- fr2en: npru-fjwu (0) < TRAMECAT (4) = Sheffield (4) < reference (11) = NLE (11) = Huawei United (11)
- it2en: Sheffield (0) < reference (4) < NLE (5) < Huawei United (7)

- pt2en: reference (2) = UNICAMP_DL (2) = Sheffield (2)
- ru2en: Sheffield (0) < TRAMECAT (3) < reference (8) = YerevaNN (8) = Huawei United (8)
- zh2en: TRAMECAT (0) < TMT (6) < Alibuba (8) < ai_not_intellegent (10) = OOM (10) = reference (10) < baidu (14) = Wei-Bot (14) = Huawei United (14)

The performance of the reference translations varied from being inferior to all runs that were validated, to being superior to all of them. However, for many language pairs, it was as good as the best runs. We summarize the performance of the reference translation below:

- Inferior to all submissions: en2es
- Superior to one or more submissions: en2de, it2en, zh2en, de2en
- Similar to the best submissions: en2it, en2pt, en2zh, pt2en, fr2en, es2en, ru2en
- Superior to all submissions: en2fr, en2ru

In general, the runs that obtained the best scores in the automatic evaluation were also the ones better ranked in the manual evaluation. We highlight the interesting differences to the automatic evaluation below:

en2es: Even though the UNICAM run obtained a slightly higher BLEU score than the ElhuyarNLP one, the latter was ranked much higher. Further, the Ixamed run was ranked reasonably high, even though it obtained the lowest BLEU score.

en2fr: The nrpu-fjwu run was ranked higher the LIMSI run, even though its BLEU score was slightly lower than the one from LIMSI.

en2zh: The run from Alibuba was ranked together with the highest runs, even though its BLEU score was the second lowest one. The Wei-Bot runs was considered as good as some other ones, even though its BLEU score was considerably higher.

de2en: While we did not observe a large difference in the BLEU scores for the runs, three teams (Huawei United, YerevaNN, NLE) were ranked higher than the other two (UNICAM and TMT).

pt2en: While the Sheffield run obtained a higher BLEU score, runs from the Sheffield and UNICAMP_DL were ranked as similar. However, as stated above, we could not perform a manual validation over a larger set of abstracts.

zh2en: The same differences that we observed for en2zh also occurred for zh2en.

es2en: Even though our evaluation relied on very few abstracts, the results confirmed the ones obtained in the automatic evaluation: all systems seem indeed to have a similar quality.

6.2 Basque abstracts

For the human evaluation of the systems that participated in the English-Basque scientific translation, we only carried out the evaluation at sentence-level. We randomly sampled a total of 100 sentences. The runs that we considered from each team are:

- en2eu (4 teams): DCU-MT (run1), Elhuyar_NLP_team (run2), Ixamed (run3), UTS_NLP (run2)

The results of the human evaluation carried out with Appraise are in Table 23, and like in the MEDLINE evaluation, bold numbers indicate a significance difference between the systems after running a Wilcoxon Signed-Rank test. The final ranking of the systems is as follows:

- en2eu: UTS_NLP (0) < DCU-MT (4) = Ixamed (4) < Elhuyar_NLP_team (10) = reference (10)

Similar to what was observed in the MEDLINE evaluation, ranking of the human evaluation matched the ranking of the automatic evaluation.

7 Discussion

In this section we present insights from the automatic and manual validations. We also reflect on the new processes introduced this year in the workflow of the task.

7.1 Analysis of results and methods

Systems submitted to the biomedical task. Figure 1 shows the correlation between BLEU and chrF scores. The use of the survey was helpful to collect specific features of the systems in order to compare the methods used. However, the variety of resources leveraged by the different teams as well as the variety of information reported about

Language	Pair	Abstracts				Sentences			
		Total	A>B	A=B	A<B	Total	A>B	A=B	A<B
en2de	TMT-Huawei	10	3	1	1	104	22	48	26
	TMT-YerevaNN		**4**	1	0		22	53	26
	TMT-NLE		3	1	1		24	62	15
	TMT-UNICAM		**4**	1	0		30	54	17
	TMT-reference		3	0	2		26	45	29
	Huawei-YerevaNN		0	4	1		13	78	11
	Huawei-NLE		2	1	2		21	57	24
	Huawei-UNICAM		1	3	1		**35**	55	12
	Huawei-reference		1	2	2		20	56	26
	YerevaNN-NLE		1	2	2		22	62	19
	YerevaNN-UNICAM		2	1	2		**29**	59	15
	YerevaNN-reference		3	0	2		21	59	23
	NLE-UNICAM		**4**	1	0		24	66	13
	NLE-reference		2	1	2		18	60	25
	UNICAM-reference		2	0	3		18	56	29
de2en	Huawei-YerevaNN	7	1	2	4	50	9	25	16
	Huawei-reference		3	1	3		13	24	13
	Huawei-UNICAM		**4**	3	0		**17**	26	7
	Huawei-TMT		2	1	4		14	27	9
	Huawei-NLE		3	3	1		10	33	7
	YerevaNN-reference		5	1	1		18	21	11
	YerevaNN-UNICAM		5	1	1		**20**	23	7
	YerevaNN-TMT		2	3	2		11	33	6
	YerevaNN-NLE		2	2	3		11	31	8
	reference-UNICAM		4	2	1		19	20	11
	reference-TMT		4	0	3		15	20	15
	reference-NLE		3	0	4		13	26	11
	UNICAM-TMT		1	1	5		8	28	14
	UNICAM-**NLE**		1	2	4		1	36	**13**
	TMT-NLE		2	1	4		5	36	9

Table 16: Manual validation for the en2de and de2en of the MEDLINE abstracts test set. The sum of the values for the sentences does not sum up to the expected value for some rows because some sentences might have been skipped. The better performing system (or reference translation) in each pairwise comparison is shown in bold, as well as the respective value that has been identified as superior.

the resources (see Table 7, 8 and 9) make it difficult to directly compare resource use in terms of type or even size. For example, some teams reported the size of their parallel datasets in terms of GB of text, some the number of aligned sentences and sometimes they provided an overall size of resources used for several language pairs.

Biomedical datasets as test suites in the news task. Overall, the best performance on the biomedical datasets was obtained by systems submitted to the biomedical task. These results suggest that domain-specific systems can offer a substantial increase in BLEU score when translating biomedical text. The performance offered by some of the news systems (e.g., Online-B, Online G) was quite high, but it has to be noted that we do not know what training data those system used, and there is no guarantee that our test sentences were not included.

We can also note that whereas no team participated both in the news and biomedical task, we submitted some of our baselines to the news task under the name *WMTbiomedBaseline*. Interestingly, our de2en baseline performed much better there (+2.5 BLEU) on the same text. This is due to supplementary processing: each paragraph to be translated was split into sentences, the sentences were translated one by one, then the results where joined back into a single paragraph. This was not done for the baseline submission to our biomedical translation task, under the assumption that the texts to translate are single-sentence (now invalidated). For the multi-sentence paragraphs, our baselines (as sent to the biomedical task) sometimes contained only the translation of the first sentence, thus leading to a decrease in BLEU score.

7.2 New additions to the workflow of the task

This year, we introduced a number of new processes into the task workflow. First, we performed manual validation of the sentence alignment for three language pairs. This resulted in higher quality alignment, and should be continued. Second, we attempted to split the test sets for the en/fr language

Language	Pair	Abstracts				Sentences			
		Total	A>B	A=B	A<B	Total	A>B	A=B	A<B
en2es	TRAMECAT-UNICAM	9	0	7	2	104	8	82	13
	TRAMECAT-Ixamed		1	7	1		8	85	10
	TRAMECAT-reference		4	2	2		13	81	10
	TRAMECAT-**ElhuyarNLP**		1	7	1		1	93	**10**
	TRAMECAT-Sheffield		3	6	0		5	90	8
	UNICAM-Ixamed		4	4	1		8	90	6
	UNICAM-reference		2	5	2		**13**	87	4
	UNICAM-ElhuyarNLP		2	7	0		4	94	6
	UNICAM-Sheffield		1	6	2		4	93	7
	Ixamed-reference		**5**	4	0		8	83	13
	Ixamed-**ElhuyarNLP**		0	5	**4**		4	93	7
	Ixamed-Sheffield		0	7	2		4	94	6
	reference-**ElhuyarNLP**		0	4	**5**		6	89	9
	reference-Sheffield		1	4	4		6	88	10
	ElhuyarNLP-Sheffield		2	6	1		4	98	2
es2en	Sheffield-Ixamed	2	2/0	0/2	0/0	14	**9**/6	5/6	0/2
	Sheffield-TRAMECAT		1/1	0/1	1/0		6/4	4/8	4/2
	Sheffield-NLE		0/9	0/2	2/0		2/1	7/11	5/2
	Sheffield-UNICAM		1/9	0/2	1/0		4/3	7/10	3/1
	Sheffield-reference		0/1	0/1	2/0		0/0	4/12	**10**/2
	Ixamed-TRAMECAT		1/1	0/0	1/1		1/2	4/8	**9**/4
	Ixamed-NLE		0/1	0/0	2/1		0/1	3/7	**11**/6
	Ixamed-UNICAM		0/0	0/0	2/2		1/0	7/7	6/**7**
	Ixamed-reference		0/1	0/0	2/1		1/2	1/7	**12**/5
	TRAMECAT-NLE		1/0	0/2	1/0		1/0	8/12	5/2
	TRAMECAT-UNICAM		0/0	0/1	2/1		3/1	5/12	6/1
	TRAMECAT-reference		0/1	0/1	2/0		0/0	5/12	**9**/2
	NLE-UNICAM		2/0	0/2	0/0		5/0	8/14	1/0
	NLE-reference		1/1	0/1	1/0		2/0	6/13	6/1
	UNICAM-reference		0/1	0/1	2/0		1/0	6/12	**7**/0

Table 17: Manual validation for the en2es and es2en of the MEDLINE abstracts test set. The sum of the values for the sentences (or abstracts) does not sum up to the expected value for some rows because some sentences (or abstracts) might have been skipped. The better performing system (or reference translation) in each pairwise comparison is depicted in bold, as well as the respective value that has been identified as superior. For es2en, two values are shown: on the left is the validation performed by the English native speaker, and on the right the one from the Spanish native speaker.

pair according to the source language as inferred from MEDLINE metadata. Our experience so far is inconclusive and shows that the initial selection of separate test sets based on source language should be done upstream in the process, as most of the test documents selected had English as the original language. The collection of system information through a survey was effective to collect general comparable information about the systems, especially as the task is growing in number of participants and language pairs offered. However, direct comparison of methods or resources is not necessarily facilitated as authors report information in different ways. A better method for yielding actionable comparisons could be to host a “constrained track" where participants would be requested to use a choice of resources provided in the track.

7.2.1 MEDLINE test sets

We previously presented (cf. Table 2) the results of the manual validation of the automatic alignment that was carried out for the test sets. Here we discuss some of the problems that we found in the automatic alignment for each of the languages.

For all the language pairs, many of the mistakes that we found referred to the titles of the articles, which are usually only available in one of the languages in MEDLINE. Therefore, many of them were correctly aligned to nothing, later identified by the evaluators as being a “NO_ALIGNMENT". However, in some cases, they was incorrectly aligned to the first sentence of the other language, which resulted in them being classified as an “OVERLAP".

The sub-sections which are present in many abstracts, such as “Background" or “Methods" were a cause for trouble. Given their simplicity, they were often correctly aligned. However, in some cases they were aligned to nothing at all (“NO_ALIGNMENT"). In other cases, they were joined to the following or previous sen-

Language	Pair	Abstracts				Sentences			
		Total	A>B	A=B	A<B	Total	A>B	A=B	A<B
en2fr	nrpu-fjwu-**Huawei**	6	0	0	**6**	100	17	19	**64**
	nrpu-fjwu-LIMSI		5	0	1		36	28	36
	nrpu-fjwu-**reference**		0	0	**6**		15	14	**71**
	nrpu-fjwu-TRAMECAT		2	2	2		38	29	33
	nrpu-fjwu-Sheffield		**4**	2	0		41	19	39
	Huawei-LIMSI		**6**	0	0		**61**	23	16
	Huawei-**reference**		1	0	5		16	8	**56**
	Huawei-TRAMECAT		**6**	0	0		**59**	33	8
	Huawei-Sheffield		**6**	0	0		**57**	32	10
	LIMSI-**reference**		0	0	**6**		9	14	**77**
	LIMSI-TRAMECAT		1	3	2		31	32	37
	LIMSI-Sheffield		1	3	2		29	27	43
	reference-TRAMECAT		**6**	0	0		**69**	25	6
	reference-Sheffield		**6**	0	0		**69**	21	8
	TRAMECAT-Sheffield		2	1	3		28	32	38
fr2en	reference-NLE	11	5	2	3	109	36	28	44
	reference-Huawei		3	1	7		37	27	43
	reference-TRAMECAT		8	1	2		**66**	26	15
	reference-Sheffield		8	0	3		**64**	20	23
	reference-nrpu-fjwu		**10**	1	0		**79**	20	8
	NLE-Huawei		5	1	5		28	57	24
	NLE-TRAMECAT		**9**	2	0		**73**	21	15
	NLE-Sheffield		**9**	1	1		**69**	29	11
	NLE-nrpu-fjwu		**11**	0	0		**89**	14	6
	Huawei-TRAMECAT		**11**	0	0		**78**	24	7
	Huawei-Sheffield		**11**	0	0		**70**	30	9
	Huawei-nrpu-fjwu		**11**	0	0		**87**	19	3
	TRAMECAT-Sheffield		4	2	5		37	29	43
	TRAMECAT-nrpu-fjwu		**8**	2	1		**64**	28	17
	Sheffield-nrpu-fjwu		8	0	3		**65**	21	23

Table 18: Manual validation for the en2fr and fr2en of the MEDLINE abstracts test set. The sum of the values for the sentences does not sum up to 109 for some rows because some sentences might have been skipped. The better performing system (or reference translation) in each pairwise comparison is depicted in bold, as well as the respective value that has been identified as superior.

Language	Pair	Abstracts				Sentences			
		Total	A>B	A=B	A<B	Total	A>B	A=B	A<B
en2it	huawei-reference	11	5	3	3	100	32	45	23
	huawei-sheffield		**10**	0	0		**80**	14	0
	reference-sheffield		**9**	0	0		**80**	12	4
it2en	sheffield-**reference**	9	0	0	**9**	100	5	1	**94**
	huawei-reference		6	1	2		**46**	37	17
	nle-reference		6	2	1		40	35	25
	huawei-sheffield		**9**	0	0		**98**	2	0
	sheffield-**nle**		0	0	**9**		1	3	**96**
	huawei-nle		3	4	2		27	53	20

Table 19: Manual validation for the en2it and it2en of the MEDLINE abstracts test set. The sum of the values for the sentences (or abstracts) does not sum up to the expected value for some rows because some sentences (or abstracts) have been skipped. The better performing system (or reference translation) in each pairwise comparison is shown in bold, as well as the respective value that has been identified as superior.

tence and aligned to a sentence in the other language, which did not contain the corresponding sub-section. Such cases were classified as either "SOURCE_GREATER_TARGET", or "TARGET_GREATER_SOURCE".

Comparing one sentence in one language that was automatic aligned to two or more sentences also sometimes caused mistakes. While most of the information is present in both languages, there were always differences between them, and more information in the language for which the alignment tool joined more than one sentence. Depending on the case, the alignment was classified as either "SOURCE_GREATER_TARGET", or "TARGET_GREATER_SOURCE".

Finally some alignments were classified as being "SOURCE_GREATER_TARGET", "TARGET_GREATER_SOURCE", or "OVERLAP"

Language	Pair	Abstracts				Sentences			
		Total	**A>B**	**A=B**	**A<B**	**Total**	**A>B**	**A=B**	**A<B**
en2pt	**reference**-UNICAMP_DL	13	**9**	3	1	107	**37**	56	14
	reference-Sheffield		6	3	4		18	69	20
	UNICAMP_DL-**Sheffield**		0	2	**11**		8	63	**36**
pt2en	reference-UNICAMP_DL	4	4/2	0/2	0/0	47	18/7	18/35	11/5
	reference-Sheffield		1/1	1/2	2/1		10/2	28/37	9/8
	UNICAMP_DL-Sheffield		0/0	1/1	3/3		6/**38**	29/9	12/0

Table 20: Manual validation for the en2pt and pt2en of the MEDLINE abstracts test set. The better performing system (or reference translation) in each pairwise comparison is shown in bold, as well as the respective value that has been identified as superior. For pt2en, two values are shown: on the left is the validation performed by the English native speaker, and on the right the one from the Portuguese native speaker.

Language	Pair	Abstracts				Sentences			
		Total	**A>B**	**A=B**	**A<B**	**Total**	**A>B**	**A=B**	**A<B**
en2ru	Huawei-**YerevaNN**	6	1	1	5	66	5	41	**18**
	Huawei-Sheffield		**5**	2	0		**29**	24	12
	Huawei-**reference**		0	2	**5**		1	40	**24**
	Huawei-TRAMECAT		3	3	1		22	31	11
	YerevaNN-Sheffield		**7**	0	0		**36**	28	1
	YerevaNN-**reference**		0	4	3		6	45	**15**
	YerevaNN-TRAMECAT		4	1	2		**26**	35	5
	Sheffield-**reference**		0	0	**7**		2	29	**35**
	Sheffield-TRAMECAT		0	5	2		10	38	18
	reference-TRAMECAT		**7**	0	0		**35**	30	1
ru2en	**Huawei**-Sheffield	6	**7**	0	0	58	**38**	15	4
	Huawei-reference		1	6	0		7	46	5
	Huawei-TRAMECAT		**6**	1	0		**24**	29	5
	Huawei-YerevaNN		2	5	0		12	40	6
	Sheffield-**reference**		0	0	**7**		1	17	**40**
	Sheffield-**TRAMECAT**		0	5	2		7	31	**20**
	Sheffield-**YerevaNN**		0	1	**6**		5	17	**34**
	reference-TRAMECAT		**4**	3	0		**19**	34	5
	reference-YerevaNN		2	4	1		9	39	10
	TRAMECAT-**YerevaNN**		0	2	**5**		8	31	18

Table 21: Manual validation for the en2ru and ru2en of the MEDLINE abstracts test set. The sum of the values for the sentences does not sum up to the expected value for some rows because some sentences might have been skipped. The better performing system (or reference translation) in each pairwise comparison is shown in bold, as well as the respective value that has been identified as superior.

when small details were present in just one of the languages. For instance, one example lacked the information about the p-value, i.e., "($p < 0.05$)", for one of the languages. For another abstract, the sentence in one language referred to the expression "in the city", while the one in the other language explicitly included the name of the city, i.e., "in Paris". It was common that a variety of small details or additional information which were not equally included for both languages.

7.2.2 Basque Abstracts test set

The alignment between the sentences for the abstracts in Basque and English was also carried out manually. Twelve sentences in Basque lack their translation in English and so these sentences in Basque were removed, resulting in the final test set of 375 pairs. The translations produced by the authors of the abstracts are not literal, and in some cases the information given in both languages is different. For example, in two consecutive sentences in an abstract about the listeriosis disease, we have these sentence pairs: First sentence (sentence 1):

- en: In recent years, we have detected a significant increase in the number of cases in Gipuzkoa.
- eu: *Azken urteotan, Gipuzkoan,* ***listeriosiaren*** *intzidentziaren igoera esanguratsua atzeman da.*
  'In recent years, in Gipuzkoa, there has been a significant increase in the incidence of listeriosis'

Following sentence (sentence 2):

- en: **Listeriosis** is uncommon in the general population, but it is far more frequent in pregnant women and newborns.

Pair	en2zh - Abstracts			Pair	zh2en - Abstracts		
	A>B	A=B	A<B		A>B	A=B	A<B
reference-TRAMECAT	**16**	1	0	**reference**-TRAMECAT	**19**	1	0
reference-baidu	9	2	4	reference-baidu	6	2	6
reference-TMT	**16**	0	1	reference-TMT	13	2	5
reference-Wei-Bot*	9	1	7	reference-Wei-Bot*	5	4	11
reference-ai_not_intellegent	9	1	7	reference-ai_not_intellegent	10	0	10
reference-OOM	8	2	7	reference-OOM	5	3	12
reference-Huawei	10	1	6	reference-Huawei	4	6	10
reference-Alibuba	7	1	6	reference-Alibuba	7	1	6
TRAMECAT-**baidu**	0	0	**14**	TRAMECAT-**baidu**	0	1	**13**
TRAMECAT-TMT	4	2	11	TRAMECAT-**TMT**	1	1	**18**
TRAMECAT-**Wei-Bot***	0	1	**16**	TRAMECAT-**Wei-Bot***	1	0	**19**
TRAMECAT-**ai_not_intellegent**	0	0	**17**	TRAMECAT-**ai_not_intellegent**	0	2	**18**
TRAMECAT-**OOM**	0	0	**17**	TRAMECAT-**OOM**	0	0	**20**
TRAMECAT-**Huawei**	0	0	**17**	TRAMECAT-**Huawei**	0	0	**20**
TRAMECAT-**Alibuba**	0	0	**14**	TRAMECAT-**Alibuba**	0	0	**14**
baidu-TMT	9	2	4	baidu-TMT	8	1	5
baidu-Wei-Bot*	0	14	1	baidu-Wei-Bot*	0	14	0
baidu-ai_not_intellegent	5	5	5	**baidu**-ai_not_intellegent	**9**	3	2
baidu-OOM	0	13	2	baidu-OOM	0	13	2
baidu-Huawei	3	7	5	baidu-Huawei	4	8	2
baidu-Alibuba	6	6	2	**baidu**-Alibuba	**9**	3	2
TMT-**Wei-Bot***	3	2	**12**	TMT-**Wei-Bot***	3	2	**15**
TMT-ai_not_intellegent	1	3	**13**	TMT-**ai_not_intellegent**	1	6	**13**
TMT-**OOM**	2	1	**14**	TMT-**OOM**	3	3	**14**
TMT-**Huawei**	2	1	**14**	TMT-**Huawei**	3	2	**15**
TMT-**Alibuba**	2	2	**11**	TMT-Alibuba	3	1	10
Wei-Bot*-ai_not_intellegent	6	7	4	**Wei-Bot***-ai_not_intellegent	**12**	6	2
Wei-Bot*-OOM	0	16	1	Wei-Bot*-OOM	0	18	2
Wei-Bot*-Huawei	6	7	4	Wei-Bot*-Huawei	7	7	6
Wei-Bot*-Alibuba	5	7	3	Wei-Bot*-Alibuba	3	3	8
ai_not_intellegent-OOM	2	7	8	ai_not_intellegent-OOM	2	5	13
ai_not_intellegent-Huawei	3	8	6	ai_not_intellegent-Huawei	4	6	10
ai_not_intellegent-Alibuba	1	13	1	ai_not_intellegent-Alibuba	0	14	0
OOM-Huawei	8	7	2	OOM-Huawei	6	11	3
OOM-Alibuba	6	6	3	OOM-Alibuba	10	1	3
Huawei-Alibuba	6	6	3	**Huawei**-Alibuba	**9**	4	1

Table 22: Manual validation for the en2zh and zh2en of the MEDLINE abstracts test set. The evaluation was carried out only for abstracts: 17 for en2zh, and 20 for zh2en. The sum of the values for the abstracts does not sum up to the expected value for some rows because some abstracts might have been skipped. The better performing system (or reference translation) in each pairwise comparison is shown in bold, as well as the number of times this system was superior. The system identified with an * cannot be fully compared to the other systems.

Pair	Sentences			
	Total	A>B	A=B	A<B
reference-UTS_NLP	100	**91**	7	2
reference-Ixamed	100	**68**	13	19
reference-Elhuyar_NLP	100	37	33	30
reference-DCU-MT	100	**75**	10	15
Ixamed-UTS_NLP	100	**60**	11	29
Ixamed-**Elhuyar_NLP**	100	17	25	**58**
Ixamed-DCU-MT	100	51	7	42
Elhuyar_NLP-UTS_NLP	100	**94**	6	0
Elhuyar_NLP-DCU-MT	100	**67**	24	9
DCU-MT-UTS_NLP	100	**74**	17	9

Table 23: Manual validation of the en2eu abstracts test set. The better performing system (or reference translation) in each pairwise comparison is shown in bold, as well as the respective value that has been identified as superior.

- eu: *Arrisku- taldeen artean, haurdun dauden emakumeak aurkitzen dira.*
  'Risk groups include pregnant women'

In the first sentence pair, the name of the disease is given in Basque, while in the second pair, the mention is given in English. In the second pair, the sentence in English gives more information than the one in Basque. This fact could well affect the

automatic evaluation.

7.3 Quality of the system translations

We discuss below some of the mistakes that we found during the manual validation of the selected runs and the reference translations.

7.3.1 MEDLINE test sets

en (from de) The quality of the translations has substantially improved since last year, with many instances requiring lengthy manual scrutiny to detect slight nuances in the meaning of the translated texts. In some cases, the subject matter of the abstracts presented a real challenge for the manual validator, as some of the translations required deeper background knowledge of medical procedures and terms to evaluate whether or not the translations from the source language were indeed correct. Examples include: (1) the German term *Hyperandrogenämie* was correctly translated to“hyperandrogenemia" (referring to elevated levels of androgen in the blood) or incorrectly to “hyperandrogenism" (refers to the state characterized by elevated levels of androgens); (2) in the context of liver cirrohsis, the “Child-Pugh-Score" was used as a pro-form term for liver cirrohsis disease severity. In this particular case, the correct translation was not even evident until the abstract was evaluated as a whole, since the manual validation of single sentences did not even contain the term *Child-Pugh-Stadium* in the source German sentence; (3) in an ophthalmology abstract, the German phrase *Aufgrund des ausgeprägten Hornhautödems* was correctly and literally translated in one instance as “Due to the pronounced corneal edema" but slightly differently in the other instance as “Due to the pronounced corneal endothelial epithelial decompensation", which may be partially correct in that corneal edema is a clinical feature of corneal endothelial epithelial decompensation. Such an interpretation would be best evaluated by an opthalmologist.

Abbreviations continue to present difficulties for correct translation. For example, in German, *Cephalosporine der 3. Generation* was never correctly translated to “third generation cephalosporins". Also the disease abbreviation *HEED (Hornhaut-Endothel-Epithel-Dekompensation)* could not be translated into English, though the disease was correctly translated in English to “corneal endothelial epithelial decompensation". The abbreviation for *polyzystische Ovarsyndrom (PCOS)* was incorrectly interpreted as a plural (“PCOs") in one translation.

Some specific medical terms were literally translated from the German source words, but resulted in an unusual or rare choice of words in English. For example, *Darm-Hirn-Achse* literally translated to “bowel-brain axis" instead of “brain-gut axis", *Adipositas* directly to “adiposity" vs. “obesity", *Mikrobiomtransfers* to “microbiome transfer" vs. “microbiota transplantation", *Kupfer-Instrauterinpessar* to “IUP" instead of “intrauterine device (IUD)". In these examples, the translations are in principle still understandable, yet awkward in English.

In some cases, choosing an English synonym of a translated German word altered the original German meaning entirely. For example, the German phrase *abgeschlossenen und laufenden kontrollierten Studien* was translated into “terminated and ongoing controlled trials" as well as “completed and ongoing controlled studies", whereby the use of the adjective “terminated" in this specific context implies that the clinical trial was prematurely stopped, possibly due to ethical, financial, safety or efficacy concerns. In this context, “completed" is the better adjective, as it implies that a study protocol was carried out to its scheduled endpoints. Similarly, in the context of raising children, the German *Erziehungserfahrungen* was sometimes translated to “educational experience", rather than the correct term “parenting experiences".

es (from en) This year, five different MT systems competed against the human reference translation for the English to Spanish language pair. The overall quality of all five systems was very good this year, when comparing sentences, being equal to the human translation in many instances.

The handling of acronyms still requires improvement for some of the MT systems, as the treatment vary from inconsistent translation, in the case of abstracts, to wrong use of lower case instead of capital letters as in the following example, correct acronym for *Sistema Único de Salud* (*SUS*) versus *Sistema Único de Salud* (*sus*). There were also some instances of literal translation of terms such as the mistranslation of *severe temperature* as *temperatura severa* when a more correct translation would have been *temperatura grave*.

In long sentences, there were also cases of missing information in the MT systems that affected the overall quality of the translations. In the rare cases where there were no clear issues with the

MT output, the human translation was sometimes more readable and more fluent and therefore the preferred choice in terms of quality. As in the following example:

- Original English text: *The objective was to assess parental knowledge, behaviors, and fears in the management of fever in their children.*
- Tramecat Translation: *El objetivo fue evaluar el conocimiento, comportamientos y miedos de los padres en el manejo de la fiebre en sus hijos.*
- Reference translation: *El objetivo fue evaluar los conocimientos, actitudes y temores de los padres ante la fiebre de sus hijos.*

The noun group elements have greater concordance in the reference translation rendering it more readable and fluent than the tramecat MT system. When comparing the reference abstracts to the MT abstracts, the human translation had higher quality due to its consistency and overall textual coherence. Some systems had issues with term translation consistency, non-fluent text (rare) or missing information (also rare). As mentioned, the MT systems performed very well when compared with one another and with the reference translation, to obtain a good level of quality, but in some cases many of the systems would still require human intervention in terms of post-edition to improve them to publishing quality level.

en (from fr) The overall quality of translations was high, with many perfect translations. Most translation issues arose from unknown vocabulary or an inappropriate use of vocabulary in context. This includes (i) the presence of untranslated French words (*We montrons* as a translation of *nous montrons* 'we show'), (ii) the erroneous translation of subword units, resulting in a merging of units (*tharural* instead of *than rural*), (iii) erroneous translation of context-dependent ambiguous terms (*Study of litter* as a translation of *étude de portée* 'scoping study' as a consequence of a poor translation of the ambiguous word *portée* 'scope, litter (of puppies)') and (iv) a strange translation of unseen source words that may nevertheless share initial subword units with the predicted word (*consumptions of cruels* as a translation of *consommation de crudités* 'consumption of raw vegetables'). A further issue noted was the poor translation of the French pronoun *il* 'it/he' into *he* when this refers to the article itself. The correct translation of these pronouns necessitates taking into account preceding context.

en (from it) The quality of the translations was neatly divided between almost-perfect and very poor, and this is reflected in the relative rankings between validations reported in Table 19. Outright errors in the good translations were rare; occasionally, the subject of a subordinate clause was mistaken. Interestingly, some translations proved capable of appropriately using synonyms and correctly rendering the meaning of the source with a slightly less literal and more idiomatic translation.

en (from zh) The quality of the translations is generally good. Some systems produced translations that provided not only correctness but also more typical English word usage beyond a literal translation. As an example, 不同性别、年龄别和身高别儿童青少年血压评价 was translated more literally by one system as *blood pressure evaluation in children and adolescents of different sexes, ages and heights*, but another system was able to produce a more natural translation: *blood pressure evaluation in children and adolescents by gender, age and height*.

The biggest source of errors is by far the translation of biomedical concepts. Presumably because a concept is not available in a reference dictionary in the target language, the translation systems often resorted to a literal interpretation of the source characters, leading to a translation that ranged from comprehensible to completely incorrect. For instance, a correct translation for 美观协调 is *aesthetic coordination* (in the context of teeth and jaw operations), but an actual and rather literal translation was *good and beautiful are in harmony*, which was still comprehensible. In another example, however, a correct translation of 早期移植物功能不全 was *early graft dysfunction*, but an incorrect translation yanked two characters 植物 (meaning "plants") out of the 3-character term 移植物 (meaning "transplant matter") and produced *early removal of plant functions*, which was completely incorrect.

A second problem area is the skipping of source words or even phrases. For biomedical texts, even skipping one critical word can significantly alter the context of the entire text. Take 老年骨质疏松人群 as an example, whose full translation is *elderly osteoporosis population*. Some translations

omitted the word *elderly*, and that changed the context of the corresponding scientific study.

fr (from en) Overall, the quality of the translations ranged from fair to good and was improved over previous editions of the task. Some aspects previously noted as difficult (e.g., co-reference, acronym definitions) were correctly translated by some of the systems at the sentence level. However, the abstract-level evaluation evidenced overall consistency issues. For example, a procedure correctly described as *cholécystectomie laparoscopique conventionnelle (CLC)* in an introductory sentence could be referred to with a different acronym , e.g., *CCC* in sentences appearing later in the same abstract. Other issues noted in previous editions remained, such as repeated portions of text (up to 96 repetitions of a word pair in one evaluated sentence) and untranslated sections, especially in passages containing complex numerical data. Some issues with technical vocabulary also led to incorrect translations. In the comparison of translation issues exhibited by different systems in the same sentence, a preference was given to medical correctness over grammatical correctness. For example, when comparing:

- Translation A: *L'étude en microscopie multiphotonique montre que,* ***comme on le attendait****, l'émiline-1* ***se colocalise*** *avec l'élastine.*

  and

- Translation B: *L'étude de microscopie multiphotonique montre que,* ***comme attendu****, l'Emiline-1* ***permet de colorer*** *avec de l'Éastine.*

where *comme attendu* (B) is grammatically preferable to *comme on le attendait* (A) as a translation of *as expected* and *se colocalise* (A) is semantically preferable to *permet de colorer* (B) as a translation of *colocalizes*, translation A is assessed as superior to translation B even though neither translation is perfect.

it (from en) The quality of the translations was strongly influenced by the systems (unknown at the time of the evaluation). Some of the translations were almost perfect and the best system was also able to use the correct technical terminology for specialized domains, such as philosophy and medicine. Other translation were partially correct, in the sense that they were understandable but with syntactic or lexical inconsistencies. For example, the term "otherness"– meaning "being different" – was incorrectly translated by the term *estraneità* (meaning "unfamiliarity") rather than the Italian equivalent *alterità*, which conveys the same meaning. Another example specific for the medical domain is the translation of the multi-word unit "visceral adhesions" by *adesivo viscerale* ("visceral sticker" as a literal translation) rather than the correct Italian equivalent *aderenze viscerali*. Finally, some other translations presented non-existent Italian words.

en (from pt) The translations have high fidelity to the source texts, but in terms of natural language style and typical word usage, the translations are clearly lacking, especially in longer sentences. There was a small number of critical errors in translating biomedical concepts, rendering the translation incomprehensible. For example, *acidentes ofídicos* was correctly translated as *snakebite* or as a more pedantic version, *snakebite envenomations*, but one incorrect translation *obscene accidents* was too obscure to hint at the original term. Lexical similarity might have been a contributing factor to errors as well. *Ofidismo* (meaning "snakebite") was translated as *ophidism* (meaning "poisoning caused by snake venom"), which was not an exact translation but still highly relevant. However, an incorrect translation *oblivinism* was, to the best of our knowledge, not an English word.

pt (from en) The translations have improved but none of the texts were perfect, since we also found mistakes in the reference translations. One of the most significant improvements, in comparison to previous years, is the lack of untranslated words; only very few of them were observed. However, one of the frequent problems still remains: poor translations of the acronyms, which are often the ones from the English (source) text. Most of the errors were actually in the small details, such as the best choice of words for a particular concept (e.g., *o processo de morte e morte* as a translation of *process of death and dying*), gender or number coordination (e.g., *na encaminhamento dos pacientes*, *programa de formação específico*), or misplacement of commas. Finally, more errors occurred in longer sentences due to their increased complexity than shorter ones, which tended to be correct.

de (from en) The overall quality of translation was high. In various cases the better translations were chosen based on small nuances, such as no capitalization errors, better ordering of words or sentence structure that sounds slightly more natural to a native speaker. Considering the original German abstracts, sentences often appeared to be freely translated, targeting an identical meaning rather than an exact translation. Therefore, in various cases, the automatic translations outperformed the reference translations, which sometimes lacked some information.

Generally the translation of acronyms appears more difficult. In multiple cases, translations used the English acronym instead of the German version, although the underlying term itself has been translated correctly. Finally we observed that some translations favored very technical terms, while others favored rather simple ones, but both correct. In those cases it is difficult to choose the better translation, if the rest of the sentences have the same quality. Generally we believe that using more complicated words does not mean that the translation of a scientific paper is necessarily better.

zh (from en) While the quality of zh2eh translations (discussed above) was already generally good, the quality of en2zh translations was generally even better in comparison.

Where applicable, a very specific term in English can be left untranslated in English in the Chinese text with good effect. Protein names such as *CD34* and long, complicated chemical names with abbreviations are prominent examples. The participating systems employed different strategies here: some repeated only the original English term, some repeated the English term as well as translated it in Chinese, and some translated it in Chinese but appended the English abbreviation.

In terms of language style, some systems produced more natural Chinese word usage than a literal translation. Take *evidence is strongest* as an example. A correct but linguistically clumsy translation was 证据最强, which means exactly "evidence is strongest." But other systems were able to produce more typical wordings such as 证据最有力 (meaning "evidence has most force") or, even better, 证据最为充分 (meaning "evidence is most sufficient").

The translation of biomedical concepts was again the biggest source of error, and again the problematic translations ranged from comprehensible to completely incorrect. For instance, *positive control* in the context of conducting experiments should be correctly translated as 阳性对照, but some system instead produced 积极的控制, which means "positively or enthusiastically take charge." Some translations were outright incorrect, such as when a simple term *fever* was translated as 百日咳, which means "whooping cough."

en (from ru) The English-Russian task was offered for the first time, with four MT systems participating and competing against the reference translation. The quality of translations were generally good, with two systems producing significantly better results. Translations frequently contained synonyms successfully carrying on the meaning of the source sentence. For example, "травматические поражения" is correctly translated as "traumatic lesions" and "traumatic injury". Observed was a range of translations, where some presented a stylistically more elegant solution then the others. For example, the phrase "reduction of pain syndrome" is better expressed as "reduce the level of pain". There was a small number of errors related to incorrect translation of biomedical key terms, resulting in translation being impractical. A mild example of incorrectly translated terminology is "spinal surgeon" instead of "spinal surgery". Skipping over segments of sentence was observed mainly in sentences with challenging tokenization.

ru (from en) The Russian-English task was offered for the first time, with four MT systems participating and competing against the reference translation. The quality of translations were generally good, with two systems producing significantly better results. Abbreviated disease names tended to cause an issue in translation. Sentences containing definition and the first mention of abbreviation contained the correct abbreviation. In subsequent sentences, the abbreviation was getting transliterated. For example, "chronic endometritis (CE)" is translated as "хроническим эндометритом (ХЭ)". However subsequent sentences refer to "CE" as "КЭ" and not as "ХЭ". Rarely observed were instances with the meaning lost in translation. For example, the source sentence "The biological age of sleep apnea patients exceeded the passport age by 41.3% and comorbid patients by 49.6%." was translated as: "Биологический возраст пациентов с апнозом сна превышал пассажиров на 41.3%, а сопутствующих на 49.6%."

7.3.2 Basque abstracts

The BLEU scores for this subtask are given in Table 14. We have to consider that BLEU scores tend to be low when translating into Basque (Jauregi Unanue et al., 2018), and this can be seen in the results. The best performing system in the automatic evaluation was Elhuyar_NLP, with a BLEU score of 0.1279. Ixamed and DCU-MT have similar performance, with UTS_NLP achieving the lowest BLEU score. In spite of the low BLEU scores, the manual evaluation in Table 23 showed that Elhuyar_NLP was competitive against the reference translation, and was preferred to other systems.

During the manual evaluation, the annotators also observed that sometimes the system produced output in Spanish instead of Basque. This was obviously a mistake when using Spanish as a pivot language, but it may have helped the BLEU scores in some cases, due to shared terminology. In the manual annotation, text in Spanish was penalized.

7.3.3 Basque terminology

As explained in Section 2.2.1, the development set and test set were the same, and this caused the results to be higher than in a real setting.[22] The results in Table 15 show that most systems performed with high accuracy and BLEU scores. Elhuyar_NLP was again the highest performer, with Ixamed producing very low scores, perhaps due to an error in their submission. We did not perform manual evaluation for this subtask.

8 Conclusions

We presented the findings of the fifth edition of the WMT biomedical task. This edition addressed three new languages and test sets that included scientific abstracts and terminologies. We explored news ways of improving our tests and carried out (as in previous editions of the task) both an automatic and a manual validation. Results confirmed the improvements of the runs and for some language pairs, suggested that some runs were on a par with or superior to the reference translations.

Acknowledgments

We would like to thank all participants in the challenges, and especially those who supported us for the manual evaluation.

[22] As a reference, one of the participating systems (UTS_NLP) was able to re-run their system over the real test set. The performance drop was 0.08 for accuracy (from 0.73 to 0.65), and 0.05 for BLEU (from 0.71 to 0.66).

References

UFAL medical corpus 1.0. https://ufal.mff.cuni.cz/ufal_medical_corpus. Accessed: 2018-07-24.

Sadaf Abdul Rauf, José Carlos Rosales Núñez, Minh Quang Pham, and François Yvon. 2020. LIMSI @ WMT 2020. In *Proceedings of the Fifth Conference on Machine Translation: Shared Task Papers*.

Rachel Bawden, Kevin Bretonnel Cohen, Cristian Grozea, Antonio Jimeno Yepes, Madeleine Kittner, Martin Krallinger, Nancy Mah, Aurelie Neveol, Mariana Neves, Felipe Soares, Amy Siu, Karin Verspoor, and Maika Vicente Navarro. 2019. Findings of the WMT 2019 biomedical translation shared task: Evaluation for MEDLINE abstracts and biomedical terminologies. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 29–53, Florence, Italy. Association for Computational Linguistics.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurelie Neveol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. 2016. Findings of the 2016 Conference on Machine Translation. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 131–198.

Ander Corral and Xabier Saralegi. 2020. Elhuyar submission to the Biomedical Translation Task 2020 on terminology and abstracts translation. In *Proceedings of the Fifth Conference on Machine Translation: Shared Task Papers*.

Ondřej Dušek, Jan Hajič, Jaroslava Hlaváčová, Jindřich Libovický, Pavel Pecina, Aleš Tamchyna, and Zdeňka Urešová. 2017. Khresmoi Summary Translation Test Data 2.0. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Christian Federmann. 2010. Appraise: An Open-Source Toolkit for Manual Phrase-Based Evaluation of Translations. In *Proceedings of the Seventh conference on International Language Resources and Evaluation*, pages 1731–1734, Valletta, Malta.

Yvette Graham, Barry Haddow, and Philipp Koehn. 2019. Translationese in machine translation evaluation. *CoRR*, abs/1906.09833.

Nicolas Griffon, Matthieu Schuers, Gaëtan Keroelhué, Julien Grosjean, and Stéfan J Darmoni. 2017. Littérature scientifique en santé (LiSSa) : une base de données bibliographiques en français [LiSSa, health scientific literature: a French bibliographic database]. *Rev Prat*, 67:134–138.

Karen Hambardzumyan, Hovhannes Tamoyan, and Hrant Khachatrian. 2020. YerevaNN's Systems for WMT20 Biomedical Translation Task: The Effect of Fixing Misaligned Sentence Pairs. In *Proceedings of the Fifth Conference on Machine Translation: Shared Task Papers*.

Inigo Jauregi Unanue, Lierni Garmendia Arratibel, Ehsan Zare Borzeshi, and Massimo Piccardi. 2018. English-Basque statistical and neural machine translation. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Inigo Jauregi Unanue and Massimo Piccardi. 2020. Pretrained Language Models and Backtranslation for English-Basque Biomedical Neural Machine Translation. In *Proceedings of the Fifth Conference on Machine Translation: Shared Task Papers*.

Antonio Jimeno Yepes, Aurelie Neveol, Mariana Neves, Karin Verspoor, Ondrej Bojar, Arthur Boyer, Cristian Grozea, Barry Haddow, Madeleine Kittner, Yvonne Lichtblau, Pavel Pecina, Roland Roller, Rudolf Rosa, Amy Siu, Philippe Thomas, and Saskia Trescher. 2017. Findings of the WMT 2017 Biomedical Translation Shared Task. In *Proceedings of the Second Conference on Machine Translation*, pages 234–247, Copenhagen, Denmark.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast Neural Machine Translation in C++. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 116–121, Melbourne, Australia.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic.

Alexandre Lopes, Rodrigo Nogueira, Roberto Lotufo, and Helio Pedrini. 2020. Lite Training Strategies for Portuguese-English and English-Portuguese Translation. In *Proceedings of the Fifth Conference on Machine Translation: Shared Task Papers*.

Nitika Mathur, Timothy Baldwin, and Trevor Cohn. 2020. Tangled up in BLEU: Reevaluating the evaluation of automatic machine translation evaluation metrics. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4984–4997, Online. Association for Computational Linguistics.

Prashant Nayak, Rejwanul Haque, and Andy Way. 2020. The ADAPT's Submissions to the WMT20 Biomedical Translation Task. In *Proceedings of the Fifth Conference on Machine Translation: Shared Task Papers*.

Sumbal Naz, Sadaf Abdul Rauf, Noor e Hira, and Sami Ul Haq. 2020. FJWU participation for the WMT20 Biomedical Translation Task. In *Proceedings of the Fifth Conference on Machine Translation: Shared Task Papers*.

Mariana Neves, Antonio Jimeno Yepes, Aurélie Névéol, Cristian Grozea, Amy Siu, Madeleine Kittner, and Karin Verspoor. 2018. Findings of the WMT 2018 Biomedical Translation Shared Task: Evaluation on MEDLINE test sets. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 324–339. Association for Computational Linguistics.

Mariana Neves, Antonio Jimeno Yepes, and Aurélie Névéol. 2016. The scielo corpus: a parallel corpus of scientific publications for biomedicine. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 2942–2948, Portorož, Slovenia. European Language Resources Association (ELRA).

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook FAIR's WMT19 news translation task submission. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 314–319, Florence, Italy. Association for Computational Linguistics.

Aurélie Névéol, Antonio Jimeno Yepes, Mariana Neves, and Karin Verspoor. 2018. Parallel Corpora for the Biomedical Domain. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA.

Wei Peng, Jianfeng Liu, Minghan Wang, Liangyou Li, Xupeng Meng, Yangm Hao, and Qun Liu. 2020. Huawei's Submissions to the WMT20 Biomedical Translation Task. In *Proceedings of the Fifth Conference on Machine Translation: Shared Task Papers*.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Danielle Saunders and Bill Byrne. 2020. Addressing Exposure Bias With Document Minimum Risk Training: Cambridge at the WMT20 Biomedical Translation Task. In *Proceedings of the Fifth Conference on Machine Translation: Shared Task Papers*.

Felipe Soares, Viviane Moreira, and Karin Becker. 2018. A large parallel corpus of full-text scientific articles. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Felipe Soares, Mark Stevenson, Diego Bartolome, and Anna Zaretskaya. 2020. ParaPat: The multi-million sentences parallel corpus of patents abstracts. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 3769–3774, Marseille, France. European Language Resources Association.

Felipe Soares and Delton Vaz. 2020. UoS Participation in the WMT20 Translation of Biomedical Abstracts. In *Proceedings of the Fifth Conference on Machine Translation: Shared Task Papers*.

Xabier Soto, Olatz Perez-de Viñaspre, Gorka Labaka, , and Maite Oronoz. 2020. Ixamed's submission description for WMT20 Biomedical shared task: benefits and limitations of using terminologies for domain adaptation. In *Proceedings of the Fifth Conference on Machine Translation: Shared Task Papers*.

Jörg Tiedemann and Santhosh Thottingal. 2020. OPUS-MT — Building open translation services for the World. In *Proceedings of the 22nd Annual Conferenec of the European Association for Machine Translation (EAMT)*, Lisbon, Portugal.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey. European Language Resources Association (ELRA).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Olatz Perez-de Viñaspre and Maite Oronoz. 2015. Snomed ct in a language isolate: an algorithm for a semiautomatic translation. In *BMC medical informatics and decision making*, volume 15, page S5. Springer.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. 2020. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272.

Xing Wang, Zhaopeng Tu, Longyue Wang, and Shuming Shi. 2020. Tencent AI Lab Machine Translation Systems for the WMT20 Biomedical Translation Task. In *Proceedings of the Fifth Conference on Machine Translation: Shared Task Papers*.

Results of the WMT20 Metrics Shared Task

Nitika Mathur
The University of Melbourne
nmathur@student.unimelb.edu.au

Johnny Tian-Zheng Wei
University of Southern California,
jwei@umass.edu

Markus Freitag
Google Research
freitag@google.com

Qingsong Ma
Tencent-CSIG,
AI Evaluation Lab
qingsong.mqs@gmail.com

Ondřej Bojar
Charles University,
MFF ÚFAL
bojar@ufal.mff.cuni.cz

Abstract

This paper presents the results of the WMT20 Metrics Shared Task. Participants were asked to score the outputs of the translation systems competing in the WMT20 News Translation Task with automatic metrics. Ten research groups submitted 27 metrics, four of which are reference-less "metrics". In addition, we computed five baseline metrics, including SENTBLEU, BLEU, TER and CHRF using the SacreBLEU scorer. All metrics were evaluated on how well they correlate at the system-, document- and segment-level with the WMT20 official human scores.

We present an extensive analysis on influence of reference translations on metric reliability, how well automatic metrics score human translations, and we also flag major discrepancies between metric and human scores when evaluating MT systems. Finally, we investigate whether we can use automatic metrics to flag incorrect human ratings.

1 Introduction

The metrics shared task[1] has been a key component of WMT since 2008, serving as a way to validate the use of automatic MT evaluation metrics and drive the development of new metrics.

We evaluate automatic metrics that score MT output by comparing them with a reference translation generated by human translators, who are instructed to translate "from scratch", without post-editing from MT. In addition, following last year's collaboration with the WMT Quality Estimation (QE) task, we also invited submissions of reference-free metrics that compare MT outputs directly with the source segment.

Similar to the last year's editions, the source, reference texts, and MT system outputs for the metric task come from the News Translation Task (Barrault et al., 2020, which we denote as Findings 2020). This year, the language pairs were English ↔ Chinese, Czech, German, Inuktitut, Japanese, Polish, Russian and Tamil. We further included systems participating in the WMT parallel corpus filtering task (Koehn et al., 2020): Khmer and Pashto to English.[2]

All metrics are evaluated based on their agreement with human evaluation. We evaluate metrics at three levels: comparing MT systems on the entire testset, segments (either sentences or short paragraphs), and new this year, documents. We introduce document-level evaluation to incentivize the development of metrics that are take into account broader context of evaluated sentences or paragraphs, following the recent emergence of document-level MT techniques.

Multiple References This year, we have two independently generated references for English ↔ German, English ↔ Russian, and Chinese → English. This lets us investigate the influence of references and the utility of multiple references. We instructed participants to score MT systems against the references individually as well as with all available references. In addition, we also supplied a set of references for English to German, that were generated by asking linguists to paraphrase the WMT reference as much as possible (Freitag et al., 2020). These references are designed to minimise translationese in the reference which could lead to metrics to be biased against systems that generate more natural text.

[1] http://www.statmt.org/wmt20/metrics-task.html

[2] Note that the metrics task inputs also included MT systems translating between German ↔ French in the News Translation Task, and English → Khmer and Pashto from the WMT parallel corpus filtering task. We are unable to evaluate metrics on these language pairs as human evaluation is not available

Proceedings of the 5th Conference on Machine Translation (WMT), pages 688–725

Evaluating Human Translations Given that we have multiple human translations, we asked participants to evaluate each human translation using the other as a reference. For these language-pairs, at least one of these human translations was included in the human evaluation, so we can directly evaluate metrics on how they rank the human translation compared to the MT systems.

Additional Human Evaluation Finally, we pose the question if some of the discrepancies between metrics and human scores can be explained by bad human ratings. We rerun some of the human evaluations by using the same template, but switching the rater pool from non-experts to professional linguists. In particular, we rerun human evaluation for a subset of translations where all metrics disagree with the WMT human evaluation. This experiment could reveal a new use case of automatic metrics and indicate that automatic metrics can be used to identify bad ratings in human evaluations.

We first give an overview of the task (Section 2) and summarize the baseline (Section 3.1) and submitted (Section 3.2) metrics. The results for system-, segment-, and document-level evaluation are provided in Sections 4, followed by a joint discussion Section 5. Section 6 describes our rerunning of human evaluation with linguists before we summarise our findings in Section 7.

We will release data, code and additional visualisations in the metrics package to be made available at http://www.statmt.org/wmt20/results.html

2 Task Setup

This year, we provided task participants with one test set for each examined language pair, i.e. a set of source texts (which are commonly ignored by MT metrics), corresponding MT outputs (these are the key inputs to be scored) and one or more reference translations.

In the system-level, metrics aim to correlate with a system's score which is an average over many human judgments of segment translation quality produced by the given system. In the segment-level, metrics aim to produce scores that correlate best with a human ranking judgment of two output translations for a given source segment. And finally, we also trial document-level evaluation this year. (more on the manual quality assessment in Section 2.3).

Segments are sentences for all language pairs except English ↔ German and Czech, and for English → Chinese, which do not contain sentence boundaries and are translated and evaluated at the paragraph-level.

Participants were free to choose which language pairs and tracks (system/segment/document and reference-based/reference-free) they wanted to take part in.

2.1 Source and Reference Texts

The source and reference texts we use are mainly sourced from this year's WMT News Translation Task (see Findings 2020).

The test set typically contains somewhere between 1000 and 2000 segments for each translation direction, with fewer segments for some paragraph-segmented test sets, and the English ↔ Inuktitut directions contain 2971 sentences.

All test sets are from the news domain, except the English ↔ Inuktitut datasets which have a mix of in-domain text from Canadian Parliament Hansards (1566 sentences) and out-of-domain news documents (1405 sentences).

We also have systems from the parallel corpus filtering task which are from the Wikipedia domain (also labelled *newstest2020* in the metrics test set). The Khmer → English and Pashto → English contain 2320 and 2719 sentences respectively.

The reference translations provided in *newstest2020* were created in the same direction as the MT systems were translating. The exceptions are English ↔ Inuktitut, Khmer → English and Pashto → English, where the testset is a mixture of "source-original" and "target-original" texts.

2.2 System Outputs

The results of the Metrics Task are affected by the actual set of MT systems participating in a given translation direction. On one hand, if all systems are very close in their translation quality, then even humans will struggle to rank them. This in turn will make the task for MT metrics very hard. On the other hand, if the task includes a wide range of systems of varying quality, correlating with humans should be generally easier. One can also expect that if the evaluated systems are of different types, they will exhibit different error patterns and various MT metrics can be differently sensitive to these patterns.

- **Parallel Corpus Filtering Task.** This task required participants to submit scores for each sentence in the provided noisy parallel texts. These scores were used to subsample sentence pairs, which was then used to train a neural machine translation system (fairseq). This was tested on a held-out subset of Wikipedia translations.

- **Regular News Tasks Systems.** These are all the other MT systems in the evaluation; differing in whether they are trained only on WMT provided data ("Constrained", or "Unconstrained") as in the previous years.

With all language pairs, in addition to the submissions to the task, the test sets also include translations from freely available web services (online MT systems), which are deemed unconstrained.

Overall, the results are based on 208 systems across 18 language pairs.

2.3 Manual Quality Assessment

Human scores were obtained using Direct Assessment, where annotators are asked to rate the adequacy of a translation compared to either the source segment or a reference translation of the same source. This year, human data was collected from reference-based evaluations (or "monolingual") and reference-free evaluations (or "bilingual"). The reference-based (monolingual) evaluations were crowdsourced, while the reference-less (bilingual) evaluations were mainly from MT researchers who committed their time to contribute to the manual evaluation for each submitted system to the translation task.

Finally, following reports that MT system translations might seem adequate when scored in isolation but not in context of the whole document, when possible, the ratings are collected for each segment with document context. Table 1 summarises the details of how human annotations were collected for various language-pairs at WMT 2020.

The English → Inuktitut dataset, which contains a mix of in-domain (Hansard) and out-of-domain (news) data, was only evaluated on out-of-domain segments, so for system level evaluation, we evaluate metric scores computed on the news domain only as well as the full test set.

See Findings 2020 for details on human evaluation.

2.3.1 System-level Golden Truth: DA

For the system-level evaluation, the collected continuous DA scores, standardized for each annotator, are averaged across all assessed segments for each MT system to produce a scalar rating for the system's performance.

The underlying set of assessed segments is different for each system. Thanks to the fact that the system-level DA score is an average over many judgments, mean scores are consistent and have been found to be reproducible (Graham et al., 2013). For more details see Findings 2020.

The score of an MT system is calculated as the average rating of the segments translated by the system.

2.3.2 Segment-level Golden Truth: DARR

Starting from Bojar et al. (2017), when WMT fully switched to DA, we had to come up with a solid golden standard for segment-level judgements. Standard DA scores are reliable only when averaged over sufficient number of judgments.[3]

Fortunately, when we have at least two DA scores for translations of the same source input, it is possible to convert those DA scores into a relative ranking judgement, if the difference in DA scores allows conclusion that one translation is better than the other. In the following, we denote these re-interpreted DA judgements as "DARR", to distinguish it clearly from the relative ranking ("RR") golden truth used in the past years.[4]

From the complete set of human assessments collected for the News Translation Task, all possible pairs of DA judgements attributed to distinct translations of the same source segment were converted into DARR better/worse judgements. Distinct translations of the same source input whose DA scores fell within 25 percentage points (which could have

[3]For segment-level evaluation, one would need to collect many manual evaluations of the exact same segment as produced by each MT system. Such a sampling would be however wasteful for the evaluation needed by WMT, so only some MT systems happen to be evaluated for a given input segment. In principle, we would like to return to DA's standard segment-level evaluation in future, where a minimum of 15 human judgements of translation quality are collected per translation and combined to get highly accurate scores for translations, but this would increase annotation costs.

[4]Since the analogue rating scale employed by DA is marked at the 0-25-50-75-100 points, we use 25 points as the minimum required difference between two system scores to produce DARR judgements. Note that we rely on judgements collected from known-reliable volunteers and crowd-sourced workers who passed DA's quality control mechanism. Any inconsistency that could arise from reliance on DA judgements collected from low quality crowd-sourcing is thus prevented.

Language pairs	source/reference	crowd/researcher	document context
iu-en	reference	crowd	No
*-en except iu-en	reference	crowd	Yes
en-*, de-fr, fr-de	source	mix of crowd and researcher*	Yes

Table 1: Direct Assessment at WMT20. Note that researcher annotations can contain some amount of professional annotations

been deemed equal quality) were omitted from the evaluation of segment-level metrics. Conversion of scores in this way produced a large set of DARR judgements for all language pairs, shown in Table 2 due to combinatorial advantage of extracting DARR judgements from all possible pairs of translations of the same source input. We see that only km-en and ps-en can suffer from insufficient number of these simulated pairwise comparisons.

The DARR judgements serve as the golden standard for segment-level evaluation in WMT19.

2.3.3 Document-level Golden Truth: DARR

As segments were scored in document context, we can compute document scores as the average human rating of the segments in the document. We acknowledge that this may be an oversimplification. First of all, we are hoping that human assessors have indicated errors in document-level coherence at at least one of the affected segments, but we have no evidence that they actually do so. Second, document-level phenomena are rather scarce and averaging segment-level scores is likely to average out these sparse observations even if they were marked at individual sentences. And lastly, in some situations, lack of cross-sentence coherence can be so critical that any strategy of composing sentence-level scores is bound to downplay the severity of the error, see e.g. Vojtěchová et al. (2019). At the current point, we have nothing better to start with but we believe that better techniques will be proposed in the future.

Graham et al. (2017) recommend around averaging 100 annotations per document to obtain reliable document scores. Since the average number of assessments we have is much less than that, we compute the ground truth in the same way as the segment level evaluation.

We first compute document scores as the average of all segment scores in the document, which we denote as DOC-DA. We then generate DOC-DARR pairs of better and worse translations of the same source document when there is at least a 25 point difference in the raw DOC-DA scores. See Table 3 for details.

In case of DARR (which we denote as DOC-DARR), all language pairs suffer from insufficient number of these simulated pairwise comparisons.

Similar to segment-level evaluation, we use the Kendall Tau-like formula (Section 2) to evaluate metric agreement with humans on the generated pairwise DARR judgements.

Note that we do not include any human-translated segments in this evaluation. In addition, iu-en is excluded from document-level evaluation because its DA judgements were collected for isolated sentences.

3 Metrics

3.1 Baselines

We agree with the call to use SacreBLEU (Post, 2018) as the standard MT evaluation scorer. We no longer report scores of the metrics from the Moses scorer, which requires tokenized text. We use the following metrics from the SacreBLEU scorer as baselines, with the default parameters:

3.1.1 SacreBLEU baselines

- BLEU (Papineni et al., 2002a) is the precision of n-grams of the MT output compared to the reference, weighted by a brevity penalty to punish overly short translations. `BLEU+case.mixed+lang.LANGPAIR+numrefs.1+smooth.exp+tok.13a+version.1.4.14`

  We run SacreBLEU with the `--sentence-score` option to obtain sentence scores for SENTBLEU; this uses the same parameters as BLEU. Although not it's intended use, we also compute system- and document-level scores for SENTBLEU as the mean segment score.

- TER (Snover et al., 2006) measures the number of edits (insertions, deletions, shifts and substitutions) required

	DA>1	Ave	DA pairs	DARR
cs-en	664	11.3	39187	14018
de-en	785	11.0	43669	16584
iu-en	2620	4.5	26120	8162
ja-en	993	9.0	36169	15193
pl-en	1001	11.8	64670	21121
ru-en	991	10.0	44664	14024
ta-en	997	7.6	26662	12789
zh-en	2000	13.8	177492	62586
km-en	1963	3.2	8295	3706
ps-en	2204	3.1	7994	3507
en-cs	1418	10.3	68587	21121
en-de	1418	6.9	30567	9339
en-iu	1268	7.9	35384	13159
en-ja	1000	9.6	41576	12830
en-pl	1000	10.6	52003	17689
en-ru	1971	5.7	28274	8330
en-ta	1000	7.9	28974	9087
en-zh	1418	10.6	72581	12652

Table 2: Segment-level: Number of judgements for DA converted to DARR data; "DA>1" is the number of source input segments in the manual evaluation where at least two translations of that same source input segment received a DA judgement; "Ave" is the average number of translations with at least one DA judgement available for the same source input segment; "DA pairs" is the number of all possible pairs of translations of the same source input resulting from "DA>1"; and "DARR" is the number of DA pairs with an absolute difference in DA scores greater than the 25 percentage point margin.

DOC-DA>1	Ave	DOC-DA pairs	DOC-DARR
cs-en 102	11.4	6041	1424
de-en 118	11.0	6579	1866
ja-en 80	8.9	2850	790
pl-en 62	11.8	4012	635
ru-en 91	9.9	4077	753
ta-en 82	7.5	2126	684
zh-en 155	13.8	13897	3085
en-cs 130	10.2	6162	1442
en-de 130	6.9	2844	669
en-iu 35	7.8	969	203
en-ja 63	9.7	2686	469
en-pl 63	10.7	3359	677
en-ru 122	5.7	1768	387
en-ta 63	7.9	1834	389
en-zh 130	10.6	6667	651

Table 3: Document-level: Number of judgements for DOC-DA converted to DOC-DARR data; "DOC-DA>1" is the number of source input documents in the manual evaluation where we have DOC-DA scores for at least two translations of that same source input documents; "Ave" is the average number of translations with at least one DOC-DA judgement available for the same source input document; "DOC-DA pairs" is the number of all possible pairs of translations of the same source input resulting from "DOC-DA>1"; and "DOC-DARR" is the number of DOC-DA pairs with an absolute difference in DOC-DA scores greater than the 25 percentage point margin.
Note that iu-en is not included as document-context was not available for this evaluation.

to transform the MT output to the reference. `TER+lang.LANGPAIR-+tok.tercom-nonorm-punct-noasian-uncased+version.1.4.14`

- CHRF (Popović, 2015) uses character n-grams instead of word n-grams to compare the MT output with the reference [5]. Version string: `chrF2+lang.LANGPAIR-+numchars.6+space.false-+version.1.4.14`.

3.1.2 CHRF++

CHRF++ (Popović, 2017) includes word unigrams and bigrams in addition to character ngrams. We ran the original Python implementation of the metric [6] with the default parameters `--ncorder 6 --nwworder 2 --beta 2`

3.2 Submissions

The rest of this section summarizes participating metrics.

3.2.1 BERT-BASE-L2, BERT-LARGE-L2, MBERT-L2

The three baselines were obtained by fine-tuning BERT (Devlin et al., 2019) on the ratings of WMT Metrics years 2015 to 2018, using a regression loss. What distinguishes the metrics is the initial BERT checkpoint: BERT-BASE-L2 uses a 12-layer Transformer architecture pre-trained on English data, MBERT-L2 is similar but trained

[5] Note that the SacreBLEU scorer does not yet implement CHRF with multiple references

[6] `chrF++.py` available at https://github.com/m-popovic/chrF

	metric	features	Learned	Scoring level seg	doc	sys	Citation/Participant	Availability
Baselines	SENTBLEU	n-grams		•	⋆	⋆	Papineni et al. (2002a)	https://github.com/mjpost/sacrebleu
	BLEU	n-grams		–	–	•	Papineni et al. (2002a)	https://github.com/mjpost/sacrebleu
	TER	edit distance		•	⊘	•	Snover et al. (2006)	https://github.com/mjpost/sacrebleu
	CHRF	character n-grams		•	⊘	•	Popović (2015)	https://github.com/mjpost/sacrebleu
	CHRF++	character n-grams		•	⊘	•	Popović (2017)	https://github.com/m-popovic/chrF
Reference-based metrics	PARBLEU	paraphrases		•	⋆	•	Univ of Edinburgh, Univ of Tartu, JHU Bawden et al. (2020)	not a public metric
	PARCHRF++	paraphrases		•	⋆	•	Univ of Edinburgh, Univ of Tartu, JHU Bawden et al. (2020)	not a public metric
	PARESIM	paraphrases	yes	•	⊘	⊘	Univ of Edinburgh, Univ of Tartu, JHU Bawden et al. (2020)	not a public metric
	PRISM	paraphrases		•	⊘	⊘	Johns Hopkins University	https://github.com/thompsonb/prism
	CHARACTER	character edit distance		•	⊘	⊘	RWTH Aachen Wang et al. (2016)	https://github.com/rwth-i6/CharacTER
	EED	character edit distance		•	⊘	⊘	RWTH Aachen Stanchev et al. (2019)	https://github.com/rwth-i6/ExtendedEditDistance
	SWSS+METEOR	semantic similarity		•	⊘	⊘	, Xu et al. (2020)	not a public metric
	MEE	word embeddings		•	⊘	⊘	IIIT - Hyderabad, Ananya Mukherjee and Sharma (2020)	not a public metric
	YISI	contextual word embeddings		•	⊘	⊘	NRC Lo (2019, 2020)	http://chikiu-jackie-lo.org/home/index.php/yisi
	BERT-BASE-L2	contextual word embeddings	yes	•	⊘	⊘	Google (Devlin et al., 2019)	(BLEURT code, private checkpoint)
	BERT-LARGE-L2,	contextual word embeddings	yes	•	⊘	⊘	Google (Devlin et al., 2019)	(BLEURT code, private checkpoint)
	MBERT-L2	contextual word embeddings	yes	•	⊘	⊘	Google (Devlin et al., 2019)	(BLEURT code, private checkpoint)
	BLEURT	contextual word embeddings	yes	•	⊘	⊘	Google (Devlin et al., 2019)	https://github.com/google-research/bleurt
	BLEURT-EXTENDED	contextual word embeddings	yes	•	⊘	⊘	Google (Devlin et al., 2019)	(BLEURT code, private checkpoint)
	YISI-COMBII	contextual word embeddings	yes	•	⊘	⊘	Google (Devlin et al., 2019)	not a public metric
	BLEURT-COMBI	contextual word embeddings	yes	•	⊘	⊘	Google (Devlin et al., 2019)	not a public metric
	COMET	predictor-estimator model	yes	•	•	⊘	Unbabel (Rei et al., 2020b)	https://github.com/Unbabel/COMET
	COMET-RANK	predictor-estimator model	yes	•	⊘	⊘	Unbabel (Rei et al., 2020b)	https://github.com/Unbabel/COMET
	COMET-HTER	predictor-estimator model	yes	•	•	⊘	Unbabel (Rei et al., 2020b)	https://github.com/Unbabel/COMET
	COMET-2R	predictor-estimator model	yes	•	⊘	⊘	Unbabel (Rei et al., 2020b)	https://github.com/Unbabel/COMET
	COMET-MQM	predictor-estimator model	yes	•	•	⊘	Unbabel (Rei et al., 2020b)	https://github.com/Unbabel/COMET
	BAQ, EQ	?	?	•	⊘	⊘	?	not a public metric
src-based	COMET-QE	predictor-estimator model	yes	•	⊘	⊘	Unbabel (Rei et al., 2020b)	https://github.com/Unbabel/COMET
	OPENKIWI-BERT	predictor-estimator model	yes	•	⊘	⊘	Unbabel Kepler et al. (2019)	https://github.com/Unbabel/OpenKiwi
	OPENKIWI-XLMR	predictor-estimator model	yes	•	⊘	⊘	Unbabel Kepler et al. (2019)	https://github.com/Unbabel/OpenKiwi
	YISI-2	contextual word embeddings		•	⊘	⊘	NRC Lo and Larkin (2020)	

Table 4: Participants of WMT20 Metrics Shared Task. "•" denotes that the metric took part in (some of the language pairs) of the segment- and/or document- and/or system-level evaluation. "⊘" indicates that the document- and system-level scores are implied, simply taking arithmetic (macro-)average of segment-level scores. "–" indicates that the metric didn't participate the track (Seg/Doc/Sys-level). "⋆" indicates that we computed the metric's document or system score for this track as the macro-average of segment scores, though the metric is not defined this way. A metric is learned if it is trained on a QE or metric evaluation dataset (i.e. pretraining or parsers don't count, but training on WMT 2019 metrics task data does).

on Wikipedia data in 102 languages, and BERT-LARGE-L2 is English-only with 24 layers.

3.2.2 BLEURT, BLEURT-EXTENDED, YISI-COMBI, BLEURT-YISI-COMBI

BLEURT (Sellam et al., 2020a) is a BERT-based regression model trained twice: first on million synthetic pairs obtained by random perturbations, then on ratings from years 2015 to 2019 of the WMT Workshop. BLEURT-EXTENDED (Sellam et al., 2020b) is a BERT-based regression model trained on human ratings of years 2015 to 2019 of the WMT Workshop, combined with BERT-Chinese for to-Chinese sentence pairs. The main checkpoint is a 24-layer Transformer, trained on a mixture of Wikipedia articles and training data from WMT Newstest in 20 languages.

YISI-COMBI: We are using YISI-1 on an mBERT model that is fine tuned on WMT data for single reference submissions. We are using aggregating internal scores in YISI over different references for the final output for multi reference submission.

BLEURT-COMBI: We are using the same output as YISI-COMBI for single reference submissions. We are mixing YISI-1, YISI-2 and BLEURT scores for different references for the multi reference submission.

3.2.3 CHARACTER

CHARACTER (Wang et al., 2016), identical to the 2016 setup, is a character-level metric inspired by the commonly applied translation edit rate (TER). It is defined as the minimum number of character edits required to adjust a hypothesis, until it completely matches the reference, normalized by the length of the hypothesis sentence. CHARACTER calculates the character-level edit distance while performing the shift edit on word level. Unlike the strict matching criterion in TER, a hypothesis word is considered to match a reference word and could be shifted, if the edit distance between them is below a threshold value. The Levenshtein distance between the reference and the shifted hypothesis sequence is computed on the character level. In addition, the lengths of hypothesis sequences instead of reference sequences are used for normalizing the edit distance, which effectively counters the issue that shorter translations normally achieve lower TER. Similarly to other character-level metrics, CHARACTER is generally applied to nontokenized outputs and references, which also holds for this year's submission with one exception. This year tokenization was carried out for en-ru hypotheses and references before calculating the scores, since this results in large improvements in terms of correlations. For other language pairs, no tokenizer was used for pre-processing.

3.2.4 COMET

COMET* metrics (Rei et al., 2020b) were build using the Estimator model or the Translation Ranking model proposed in Rei et al. (2020a). Those neural models use XLM-RoBERTa to encode source, MT hypothesis and reference in the same cross-lingual space and then are optimised towards different objectives. COMET (main metric) is an Estimator model that regresses on Direct Assessments (DA) from 2017 to 2019 and COMET-2R is a variant of COMET (main metric) that was trained to handle multiple references at inference time. COMET-HTER and COMET-MQM follow the same architecture but regress on Human-mediated Translation Edit Rate (HTER) and a proprietary metric compliant with the Multidimensional Quality Metrics framework (MQM), respectively. COMET-Rank uses the Translation Ranking architecture to directly optimize the distance between "better" hypothesis and the respective source and reference, while pushing the "worse" hypothesis away. This Translation Ranking model was directly optimised on DA relative-ranks from 2017 to 2019. Finally, COMET-QE removes the reference at input and proportionately reduces the dimensions of the estimator network to accommodate the reduced input.

3.2.5 EED

EED (Stanchev et al., 2019) is a character-based metric, which builds upon CDER. It is defined as the minimum number of operations of an extension to the conventional edit distance containing a "jump" operation. The edit distance operations (insertions, deletions and substitutions) are performed at the character level and jumps are performed when a blank space is reached. Furthermore, the coverage of multiple characters in the hypothesis is penalised by the introduction of a coverage penalty. The sum of the length of the reference and the coverage penalty is used as the normalisation term.

3.2.6 MEE

MEE (Ananya Mukherjee and Sharma, 2020) is an automatic evaluation metric that leverages the similarity between embeddings of words in candi-

date and reference sentences to assess translation quality. Unigrams are matched based on their surface forms, root forms and meanings which aids to capture lexical, morphological and semantic equivalence. Semantic evaluation is achieved by using pretrained fasttext embeddings provided by Facebook to calculate the word similarity score between the candidate and the reference words. MEE computes evaluation score using three modules namely exact match, root match and synonym match. In each module, fmean-score is calculated using harmonic mean of precision and recall by assigning more weight to recall. The final translation score is obtained by taking average of fmean-scores from individual modules.

3.2.7 ESIM

Enhanced Sequential Inference Model (Chen et al., 2017) is a neural model proposed for Natural Language Inference that has been adapted for MT evaluation by Mathur et al. (2019). It uses cross-sentence attention and sentence matching heuristics to generate a representation of the translation and the reference, which is fed to a feedforward regressor. This year's scores were submitted by Bawden et al. (2020) as part of the submission on PARESIM.

3.3 OPENKIWI-BERT, OPENKIWI-XLMR

OPENKIWI-BERT and OPENKIWI-XLMR (Kepler et al., 2019) are state of the art Quality Estimation models developed for the WMT20 QE shared task and are trained with WMT Metrics data from 2017 to 2019.

3.3.1 PARBLEU, PARCHRF++, PARESIM

PARBLEU, PARCHRF++, and PARESIM (Bawden et al., 2020) are variants of their respective core metrics computed against the provided human reference and a set of automatically generated paraphrases. PARBLEU used five paraphrases, while the other two used only one. Both BLEU and CHRF++ have in-built support for multiple references. For ESIM, we calculate the score for each reference separately and then average them to get the final score.

3.3.2 PRISM

PRISM (Thompson and Post, 2020) is a many-many multilingual neural machine translation system trained on data for 39 language pairs, with data derived largely from WMT and Wikimatrix. It casts machine translation evaluation as a zero-shot paraphrasing task, producing segment-level scores by force-decoding between a system output and a reference, in both directions, and averaging the model scores. System-level scores are produced by averaging segment-level ones. For evaluation in Inuktikut, Khmer, Pashto, and Tamil, we used a "Prism44" model that was retrained after adding WMT-provided data for these languages to its original training data set. All other languages were evaluated with the original "Prism39" model.

3.3.3 SWSS+METEOR

SWSS (Semantically Weighted Sentence Similarity, Xu et al. 2020) is an approach to extracting semantic core words, which are words that carry important semantic meanings in sentences, and using them in MT evaluation. It uses UCCA (Universal Conceptual Cognitive Annotation), a semantic representation framework, to identify semantic core words, and then calculates sentence similarity scores on the overlap of semantic core words of sentence pairs. Taking sentence-level semantic structure information into consideration, SWSS can improve the performance of lexical metrics when combined with them. The submitted metric (SWSS+METEOR) is a weighted combination of SWSS and Meteor.

3.3.4 YISI-0, YISI-1, YISI-2

YISI (Lo, 2019, 2020) is a unified semantic MT quality evaluation and estimation metric for languages with different levels of available re-sources. YISI-1 is a reference-based MT evaluation metric that measures the semantic similarity between a ma-chine translation and human references by aggregating the idf-weighted lexical semantic similarities based on the contextual embeddings extracted from pretrained language models (BERT, CamemBERT, RoBERTa, XLM, XLM-RoBERTa, etc.) and optionally incorporating shallow semantic structures (denoted as YISI-1_SRL; not participating this year). YISI-0 is the degenerate version of YISI-1 that is ready-to-deploy to any language. It uses longest common character substring to measure the lexical similarity. YISI-2 (Lo and Larkin, 2020) is the bilingual, reference-less version for MT quality estimation, which uses bilingual mappings of the contextual embeddings extracted from pretrained language models (XLM or XLM-RoBERTa) to evaluate the crosslingual lexical semantic similarity between the input and

MT output. Like YISI-1, YISI-2 can exploit shallow semantic structures as well (denoted as YISI-2_SRL; does not participate this year).

3.4 Pre-processing

Since some metrics, such as BLEU, aim to achieve a strong positive correlation with human assessment, while error metrics, such as TER, aim for a strong negative correlation, in previous years we compare metrics via the absolute value $|r|$ of a given metric's correlation with human assessment. However, this can mask instances of true negative correlation for metrics that aim for a positive correlation (and vice-versa).

For system, document and segment level scores, we reverse the sign of the score of error metrics prior to the comparison with human scores, whether on the system, document or segment level: higher scores have to indicate better translation quality.

4 Results

4.1 System-Level Evaluation

As in previous years, we employ the Pearson correlation (r) as the main evaluation measure for system-level metrics. The Pearson correlation is as follows:

$$r = \frac{\sum_{i=1}^{n}(H_i - \overline{H})(M_i - \overline{M})}{\sqrt{\sum_{i=1}^{n}(H_i - \overline{H})^2}\sqrt{\sum_{i=1}^{n}(M_i - \overline{M})^2}} \quad (1)$$

where H_i are human assessment scores of all systems in a given translation direction, M_i are the corresponding scores as predicted by a given metric. $\overline{H}$ and $\overline{M}$ are their means, respectively.

As recommended by Graham and Baldwin (2014), we employ Williams significance test (Williams, 1959) to identify differences in correlation that are statistically significant. Williams test is a test of significance of a difference in dependent correlations and therefore suitable for evaluation of metrics. Correlations not significantly outperformed by any other metric for the given language pair are highlighted in bold in all the results tables that show Pearson correlation of metric and human scores.

Pearson correlation is ideal for reporting whether metric scores have the same trend as human scores. In practice, we use metrics to make decisions comparing MT systems, and Kendall's Tau appears to be more close to this use case, as it directly checks whether the metric ordering of a pair of MT systems agrees with the human ordering. However, unlike Pearson correlation, it is not sensitive to whether the metric score differences correspond to the human score differences. We stay with Pearson correlation for the official results, but also report Kendall's Tau correlation in the appendix.

The calculation of Pearson correlation coefficient is dependent on the mean, which is very sensitive to outliers. So if we have systems whose scores are far away from the rest of the systems, the presence of these "outlier" systems can give a misleadingly high impression of the correlations, and potentially change ranking of metrics. To avoid this, we also report correlations over non-outlier systems only.

To remove outliers, we are guided by the robust outlier detection method proposed for MT metric evaluation by Mathur et al. (2020). This method, recommended by the statistics literature (Iglewicz and Hoaglin, 1993; Rousseeuw and Hubert, 2011; Leys et al., 2013) depends on the median and the median absolute deviation (MAD) which is the median of the absolute difference between each point and the median. The method removes systems whose human scores are greater than 2.5 MAD away from the median.

The cutoff of 2.5 is subjective, and Leys et al. (2013) suggest the guidelines of using 3 (very conservative), 2.5 (moderately conservative) or 2 (poorly conservative), and recommends 2.5. For some language pairs, we override the 2.5 cutoff for systems that are close to the cutoff. We give examples in Section 5, and list all identified outliers in Table 15 in the Appendix.

4.1.1 System-Level Results

Tables 5 and 6 provide the system-level correlations of metrics. These tables include results for all MT systems, and in cases where we detect outliers, we also report correlation without outliers.

This year, we also carry out an extended analysis of the impact of (multiple) human references, see the following paragraphs.

Scoring Human Translation In this section, we investigate how well the metric submissions score human translations. We have five language pairs where two reference translations were provided by WMT. The manual DA scoring of News Translation Task included all the out-of-English human references in the evaluation along with the MT systems.

	cs-en 10		de-en 9		ja-en 7		pl-en 13		ru-en 10		ta-en 12		zh-en 15		iu-en 9		km-en 7	ps-en 6
SENTBLEU	0.844	**0.800**	0.978	**0.786**	**0.974**	**0.851**	**0.502**	**0.284**	**0.916**	**0.833**	0.925	**0.829**	0.948	**0.950**	0.649	**0.469**	0.969	**0.888**
BLEU	**0.851**	**0.800**	0.985	0.778	**0.969**	0.826	**0.549**	**0.355**	0.884	0.761	0.916	**0.807**	0.956	**0.957**	0.569	0.348	0.969	**0.888**
TER	**0.845**	**0.783**	0.993	**0.766**	**0.974**	0.752	**0.586**	**0.346**	**0.904**	**0.829**	0.805	**0.795**	0.956	0.911	**0.733**	**0.616**	**0.973**	**0.935**
CHRF++	**0.867**	**0.804**	0.997	0.699	**0.974**	**0.871**	**0.538**	**0.328**	0.894	**0.833**	**0.953**	**0.830**	**0.975**	**0.955**	0.726	0.392	0.983	0.900
CHRF	**0.872**	**0.806**	**0.997**	0.687	0.968	**0.861**	**0.528**	**0.312**	0.890	0.831	**0.951**	**0.828**	**0.976**	**0.954**	0.729	0.337	0.978	0.898
PARBLEU	**0.834**	**0.774**	0.986	**0.838**	**0.970**	**0.833**	**0.562**	**0.342**	0.877	0.744	0.908	**0.801**	0.958	**0.953**	0.624	0.398	**0.971**	**0.939**
PARCHRF++	**0.865**	**0.810**	**0.998**	0.708	**0.974**	**0.877**	**0.551**	**0.347**	0.885	0.823	**0.942**	**0.825**	**0.976**	**0.956**	0.720	0.296	**0.985**	0.899
CHARACTER	0.844	**0.812**	**0.998**	0.687	**0.970**	**0.895**	**0.522**	**0.325**	**0.927**	**0.869**	**0.965**	**0.880**	0.964	**0.950**	**0.763**	**0.410**	**0.977**	0.841
EED	**0.884**	**0.838**	**0.997**	**0.752**	**0.974**	**0.904**	**0.538**	**0.299**	**0.926**	**0.872**	**0.958**	**0.862**	0.956	0.932	**0.821**	**0.587**	**0.990**	**0.930**
YISI-0	**0.876**	**0.825**	**0.998**	**0.786**	**0.972**	**0.867**	0.453	**0.207**	**0.938**	**0.874**	**0.968**	**0.861**	0.956	0.918	**0.831**	**0.563**	**0.986**	**0.932**
SWSS+METEOR	–		–		**0.978**	**0.919**	**0.472**	**0.212**	**0.925**	**0.876**	**0.967**	**0.862**	0.959	0.926	**0.766**	**0.545**	**0.990**	**0.946**
MEE	**0.861**	**0.822**	0.995	0.712	**0.982**	**0.900**	**0.464**	**0.295**	**0.927**	**0.878**	**0.950**	**0.835**	0.952	**0.948**	**0.771**	**0.562**	0.970	0.878
PRISM	0.818	0.720	**0.998**	**0.775**	**0.974**	**0.869**	**0.502**	**0.269**	**0.908**	**0.839**	0.898	**0.788**	0.957	**0.945**	**0.833**	**0.616**	0.950	**0.966**
YISI-1	**0.832**	**0.746**	**0.998**	**0.783**	**0.982**	**0.868**	**0.543**	**0.316**	**0.915**	**0.833**	0.925	**0.797**	0.961	**0.942**	**0.834**	**0.590**	**0.977**	**0.953**
BERT-BASE-L2	0.775	0.693	0.997	0.791	0.971	0.789	**0.552**	**0.328**	0.919	**0.836**	0.909	0.746	**0.967**	0.929	0.704	0.145	0.967	**0.945**
BERT-LARGE-L2	**0.784**	0.695	0.990	0.800	**0.974**	**0.784**	0.520	0.282	**0.925**	**0.843**	0.901	0.760	0.962	0.928	0.744	0.211	0.959	**0.950**
MBERT-L2	**0.798**	0.715	0.995	**0.824**	**0.969**	**0.811**	**0.555**	**0.302**	**0.908**	0.805	0.887	0.740	0.959	**0.935**	**0.837**	**0.530**	**0.980**	**0.938**
BLEURT	**0.792**	0.725	0.996	0.770	**0.978**	**0.820**	**0.591**	**0.371**	**0.924**	**0.844**	0.906	0.768	**0.966**	**0.931**	0.771	0.320	**0.984**	**0.955**
BLEURT-EXTENDED	0.771	0.668	0.985	**0.818**	0.961	**0.772**	**0.551**	0.298	0.900	0.797	0.897	0.743	0.945	0.931	0.789	0.359	**0.985**	**0.942**
ESIM	**0.790**	0.716	**0.998**	0.808	**0.983**	**0.822**	**0.591**	**0.358**	**0.928**	**0.834**	0.885	**0.801**	**0.963**	0.910	**0.807**	**0.514**	0.929	0.929
PARESIM-1	**0.788**	0.712	**0.998**	**0.835**	**0.983**	**0.819**	**0.591**	**0.363**	**0.926**	**0.828**	0.885	**0.797**	**0.963**	0.910	**0.800**	**0.495**	0.929	0.929
COMET	0.783	0.694	**0.998**	0.773	0.964	**0.828**	**0.591**	**0.345**	**0.923**	**0.836**	0.880	**0.764**	0.952	**0.931**	**0.852**	**0.605**	0.971	0.941
COMET-2R	0.777	0.697	**0.998**	0.772	0.964	**0.818**	**0.584**	**0.332**	**0.924**	**0.843**	0.881	**0.770**	0.949	**0.928**	**0.872**	**0.644**	0.970	**0.949**
COMET-HTER	0.738	0.661	0.995	**0.767**	0.912	**0.702**	0.446	0.231	0.867	0.741	0.726	0.595	0.809	0.873	**0.770**	**0.464**	0.901	0.862
COMET-MQM	0.728	0.612	0.991	**0.684**	0.906	0.707	0.424	0.222	0.858	0.746	0.767	0.617	0.784	0.862	**0.841**	**0.631**	0.914	0.880
COMET-RANK	0.705	0.534	0.964	**0.757**	0.923	**0.793**	0.483	**0.284**	0.868	0.732	0.787	0.664	0.877	**0.909**	0.158	0.214	0.911	0.855
BAQ_DYN	–		–		–		–		–		–		0.956	0.928	–		–	–
BAQ_STATIC	–		–		–		–		–		–		**0.960**	**0.933**	–		–	–
COMET-QE	0.755	0.622	0.939	**0.805**	0.892	**0.585**	0.447	0.218	0.883	0.773	0.795	0.672	0.847	0.887	**0.685**	**0.661**	0.896	0.832
OPENKIWI-BERT	0.726	**0.698**	0.989	**0.741**	0.735	0.546	0.355	0.187	**0.862**	0.695	0.645	0.469	0.625	0.774	-0.126	-0.671	0.751	0.753
OPENKIWI-XLMR	0.760	**0.680**	0.995	0.701	0.931	0.714	0.442	0.171	0.859	0.697	0.792	0.659	0.905	0.899	0.271	-0.577	0.880	0.865
YISI-2	0.764	0.640	0.988	0.404	**0.971**	**0.776**	**0.437**	**0.230**	**0.825**	**0.814**	0.849	**0.761**	**0.964**	0.933	**0.676**	**0.371**	0.790	**0.942**

Table 5: Pearson correlation of to-English system-level metrics with DA human assessment over MT systems using the *newstest2020* references. For language pairs that contain outlier systems, we also show correlation after removing outlier systems ("-out"). Correlations of metrics not significantly outperformed by any other for that language pair are highlighted in bold.

	en-cs		en-de		en-ja		en-pl		en-ru	en-ta		en-zh	en-iu$_{\text{FULL}}$		en-iu$_{\text{NEWS\#}}$	
	All	-out	All	-out	All	-out	All	-out	All	All	-out	All	All	-out	All	-out
	12	10	14	11	11	9	14	11	9	15	12	12	11	8	11	8
sentBLEU	0.840	0.436	0.934	0.823	0.946	**0.976**	0.950	**0.772**	**0.981**	0.881	0.852	0.927	0.129	0.047	0.075	**0.172**
BLEU	0.825	0.390	0.928	0.825	0.945	**0.980**	0.943	**0.743**	**0.980**	0.880	0.829	0.928	0.163	0.131	0.074	**0.111**
TER	0.814	0.339	0.941	**0.848**	0.297	0.801	0.893	0.553	0.064	0.870	**0.883**	-0.213	0.384	**0.133**	0.357	**0.083**
chrF++	0.833	0.349	0.958	**0.850**	0.952	0.945	**0.956**	**0.783**	**0.983**	0.929	**0.880**	0.878	0.328	0.128	0.315	**0.098**
chrF	0.826	0.313	**0.962**	**0.862**	0.951	0.964	**0.957**	**0.793**	**0.982**	0.937	**0.890**	0.923	0.350	**0.122**	0.336	**0.091**
parBLEU	0.870	0.543	0.910	0.774	0.869	0.813	0.948	**0.760**	**0.959**	0.871	**0.849**	0.962	0.194	**0.464**	0.126	**0.306**
parchrF++	0.860	0.438	0.957	**0.845**	0.955	0.951	**0.953**	**0.818**	**0.975**	–		0.948	–		–	
CharacTER	0.807	0.269	**0.961**	**0.868**	0.951	**0.936**	0.935	**0.726**	0.961	0.957	0.851	0.905	0.503	0.008	0.515	**0.121**
EED	0.817	0.271	**0.965**	**0.869**	0.955	0.965	**0.962**	**0.789**	**0.980**	**0.959**	**0.913**	0.928	0.519	0.043	0.483	**0.122**
MEE	0.875	0.495	0.954	0.820	–		**0.952**	**0.733**	0.724	0.906	0.861	–	0.287	0.094	0.242	**0.113**
YiSi-0	0.797	0.270	0.953	**0.889**	0.967	**0.972**	**0.953**	**0.783**	**0.971**	0.929	**0.897**	0.362	0.525	0.015	0.505	**0.095**
prism	0.949	0.805	0.958	**0.851**	0.932	0.921	**0.958**	**0.742**	0.724	0.863	0.452	0.221	**0.957**	**-0.043**	**0.945**	**0.088**
YiSi-1	0.922	0.664	**0.971**	**0.887**	0.969	0.967	**0.964**	**0.714**	0.926	**0.973**	**0.909**	0.959	0.554	-0.217	0.523	-0.014
YiSi-combi	–		0.971	0.868	–		–		–	–		–	–		–	
bleurt-YiSi-combi	–		0.971	0.868	–		–		–	–		–	–		–	
mBERT-L2	0.946	0.782	**0.970**	**0.861**	0.977	**0.969**	**0.976**	**0.775**	0.946	**0.944**	0.834	0.934	–		–	
BLEURT-extended	**0.989**	**0.960**	**0.969**	**0.870**	0.944	0.953	**0.982**	**0.828**	**0.980**	0.940	0.814	0.928	0.823	**0.122**	0.762	**0.155**
esim	0.908	0.575	**0.979**	**0.894**	**0.993**	**0.981**	**0.969**	0.698	**0.967**	0.937	0.833	0.972	0.814	**0.365**	0.760	0.418
paresim-1	0.919	0.635	**0.974**	**0.886**	**0.989**	**0.971**	**0.968**	0.705	**0.964**	0.937	0.833	**0.983**	0.814	**0.365**	0.760	0.418
COMET	0.978	0.926	**0.972**	**0.863**	0.974	**0.969**	**0.981**	**0.800**	0.925	0.944	0.798	0.007	0.860	0.028	**0.858**	0.152
COMET-2R	**0.983**	0.942	**0.972**	**0.869**	**0.986**	**0.978**	**0.982**	**0.803**	0.872	**0.959**	0.852	-0.066	0.848	-0.008	**0.867**	**0.177**
COMET-HTER	0.976	0.917	0.951	**0.852**	**0.989**	**0.974**	**0.974**	**0.763**	0.803	0.925	0.681	-0.073	**0.900**	0.142	**0.888**	0.092
COMET-MQM	0.974	0.910	0.881	0.840	0.974	**0.965**	0.967	**0.766**	0.788	0.910	0.641	0.084	0.870	0.129	**0.867**	0.172
COMET-Rank	0.959	0.868	0.877	**0.860**	0.931	0.928	0.957	**0.760**	0.676	0.876	0.511	0.540	0.283	0.099	0.392	**0.252**
BAQ_dyn	–		–		–		–		–	–		0.904	–		–	
BAQ_static	–		–		–		–		–	–		**0.958**	–		–	
EQ_dyn	–		–		–		–		–	–		0.948	–		–	
EQ_static	–		–		–		–		–	–		**0.976**	–		–	
COMET-QE	**0.989**	**0.974**	0.903	0.831	0.953	**0.955**	0.969	**0.804**	0.807	0.887	0.622	0.375	**0.905**	**0.578**	**0.928**	**0.651**
OpenKiwi-Bert	0.920	0.830	0.852	**0.829**	0.363	0.783	0.903	0.450	0.834	0.846	0.370	0.551	0.573	-0.602	0.808	**0.194**
OpenKiwi-XLMR	0.972	0.911	**0.968**	0.814	**0.992**	**0.976**	0.957	**0.638**	0.875	0.910	0.676	-0.010	0.513	-0.668	0.680	-0.358
YiSi-2	0.714	0.353	0.899	0.552	0.854	0.646	0.470	-0.107	0.584	0.922	**0.923**	-0.215	0.802	**-0.257**	**0.830**	**0.065**

Table 6: Pearson correlation of out-of-English system-level metrics with DA human assessment over MT systems using the newstest2020 references; For language pairs that contain outlier systems, we also show correlation after removing outlier systems. Correlations of metrics not significantly outperformed by any other for that language pair are highlighted in bold.

The English → Inuktitut human evaluation only contained the news subset, so we recompute en-iu system scores of metrics on the news subset of the testset (1405 sentences).

Note that the scores of parBLEU and parchrF were computed as average of segment scores

HID N	en-de Human-B 12	en-de$_P$ Human-B 12	en-de$_B$ Human-A 12	en-de$_P$ Human-A 12	en-zh Human-B 13	en-zh$_B$ Human-A 13	de-en Human-B 10	ru-en Human-B 11	zh-en Human-B 16
SENTBLEU	0.441	0.851	0.639	0.676	0.647	0.837	0.437	0.797	0.917
BLEU	0.458	**0.868**	0.672	0.665	0.658	0.814	0.480	0.738	**0.938**
TER	0.233	0.495	0.577	0.695	-0.131	-0.138	0.466	**0.812**	0.850
CHRF++	0.555	**0.917**	0.748	0.650	0.592	0.805	0.437	0.815	**0.947**
CHRF	0.599	**0.919**	0.772	0.645	0.651	0.812	0.442	0.821	**0.948**
PARBLEU	0.349	0.676	0.580	0.682	0.569	0.787	0.498	0.716	0.926
PARCHRF++	0.573	**0.890**	0.748	0.698	0.559	0.776	0.447	0.803	**0.950**
CHARACTER	0.472	**0.890**	0.736	0.638	0.687	0.850	0.410	**0.856**	**0.938**
EED	0.447	**0.898**	0.685	0.646	0.679	0.830	0.466	**0.861**	0.910
YISI-0	0.514	**0.892**	0.728	0.724	0.244	0.274	0.566	**0.860**	0.898
SWSS+METEOR	–	–	–	–	–	–	–	**0.866**	0.914
MEE	0.512	**0.886**	0.719	0.642	–	–	0.399	**0.855**	**0.941**
PRISM	0.472	0.727	0.731	0.742	0.157	0.166	0.591	**0.837**	**0.942**
YISI-1	0.640	**0.895**	**0.830**	0.697	0.773	**0.916**	**0.713**	**0.822**	**0.943**
YISI-COMBI	0.607	0.891	0.801	0.702	–	–	–	–	–
BLEURT-YISI-COMBI	0.607	0.891	0.801	0.702	–	–	–	–	–
BERT-BASE-L2	–	–	–	–	–	–	**0.785**	**0.813**	**0.922**
BERT-LARGE-L2	–	–	–	–	–	–	**0.794**	**0.819**	**0.923**
MBERT-L2	**0.845**	0.876	**0.875**	**0.810**	0.868	**0.907**	**0.748**	0.789	**0.925**
BLEURT	–	–	–	–	–	–	0.754	**0.823**	**0.923**
BLEURT-EXTENDED	**0.888**	**0.896**	**0.883**	**0.838**	0.865	**0.910**	**0.811**	0.757	0.914
ESIM	0.719	**0.920**	**0.870**	**0.744**	0.837	**0.924**	0.765	**0.819**	**0.911**
PARESIM-1	0.687	**0.905**	**0.856**	**0.763**	0.822	**0.910**	**0.798**	**0.815**	**0.911**
COMET	0.854	**0.894**	**0.879**	**0.822**	0.078	0.062	**0.759**	**0.821**	**0.916**
COMET-2R	0.820	**0.866**	**0.877**	**0.865**	0.009	-0.003	**0.756**	**0.837**	**0.911**
COMET-HTER	0.840	0.871	**0.869**	**0.851**	0.006	-0.001	**0.761**	0.718	0.857
COMET-MQM	0.839	0.876	0.859	0.825	0.158	0.154	**0.682**	0.722	0.846
COMET-RANK	0.782	**0.870**	**0.830**	0.794	0.578	0.565	**0.709**	0.725	**0.896**
BAQ_DYN	–	–	–	–	0.739	–	–	–	0.236
BAQ_STATIC	–	–	–	–	**0.915**	–	–	–	0.239
EQ_DYN	–	–	–	–	0.729	–	–	–	–
EQ_STATIC	–	–	–	–	**0.925**	–	–	–	–
COMET-QE	**0.885**	**0.885**	**0.844**	**0.844**	0.473	0.481	**0.806**	0.749	0.865
OPENKIWI-BERT	0.741	0.741	**0.835**	**0.835**	0.487	0.521	**0.655**	0.682	0.742
OPENKIWI-XLMR	0.736	0.736	0.795	**0.795**	0.053	0.050	0.660	0.694	**0.893**
YISI-2	-0.333	-0.333	-0.039	-0.039	-0.190	-0.198	0.123	0.513	0.882

Table 7: Evaluating Human translation: Pearson correlation of metrics with DA human assessment for all MT systems plus Human translation. The subscript *B* represents an alternate reference, *P* represents a paraphrased reference. N is the total number of MT systems (excluding outliers) and HID is the identity of the human translation evaluated. Correlations of metrics not significantly outperformed by any other for that language pair are highlighted in bold.

For to-English language pairs, only the secondary human reference translations were manually scored with DA as the primary human reference translation was shown to the monolingual annotators.

For these language pairs, the metrics can score a human translation by using the other one as the reference translation. For simplicity, we add the second human reference translation to the list of translation outputs and observe how its scoring by the given metric affects the correlation.

Table 7 shows how well the metrics correlate with the WMT human evaluation when including human translations as additional output. In most cases, the correlation decreases as metrics struggle to correctly score translations that are different from MT systems. Metrics that rely on fine-tuning on existing human assessments from the previous WMT campaigns (e.g. BLEURT, ESIM, COMET) can handle human translations much better on average. Also, the Paraphrased references help the lexical metrics correctly identify the high quality of human translations.

We present a deeper analysis of how metrics score human translations in Section 5.1.2. We base this discussion on scatterplots of human vs metric scores. We include scatterplots of selected metrics in Appendix B.

Influence of References Rewarding multiple alternative translations is the primary motivation behind multiple-reference based evaluation. It is generally assumed that using multiple reference translation for automatic evaluation is helpful as we cover a wider space of possible translations (Papineni et al., 2002b; Dreyer and Marcu, 2012; Bojar et al., 2013). Nevertheless, new studies (Freitag et al., 2020) showed that multi-reference evaluation does not improve the correlation for high quality output anymore. Since we have multiple references available for five language pairs, we can look at how much the choice of reference(s) influences correlation.

Table 8 compares metric correlations on the primary reference set *newstest2020*, alternative reference *newstestB2020*, paraphrased reference *newstestP2020* (only for English-German), or using all available references *newstestM2020*. We only report system-level correlations of metrics on MT systems after discarding outliers.

4.2 Segment- and Document-Level Evaluation

Segment-level evaluation relies on the manual judgements collected in the News Translation Task evaluation. This year, again we were unable to follow the methodology outlined in Graham et al. (2015) for evaluating of segment-level metrics because the sampling of segments did not provide sufficient number of assessments of the same segment. We therefore convert pairs of DA scores for competing translations to DARR better/worse preferences as described in Section 2.3.2. We further follow the same process to generate DARR ground truth for documents, as we do not have enough annotations to obtain accurate human scores.

We measure the quality of metrics' scores against the DARR golden truth using a Kendall's Tau-like formulation, which is an adaptation of the conventional Kendall's Tau coefficient. Since we do not have a total order ranking of all translations, it is not possible to apply conventional Kendall's Tau given the current DARR human evaluation setup (Graham et al., 2015).

Our Kendall's Tau-like formulation, τ, is as follows:

$$\tau = \frac{|Concordant| - |Discordant|}{|Concordant| + |Discordant|} \quad (2)$$

where $Concordant$ is the set of all human comparisons for which a given metric suggests the same order and $Discordant$ is the set of all human comparisons for which a given metric disagrees. The formula is not specific with respect to ties, i.e. cases where the annotation says that the two outputs are equally good.

The way in which ties (both in human and metric judgement) were incorporated in computing Kendall τ has changed across the years of WMT Metrics Tasks. Here we adopt the version used in WMT17 DARR evaluation. For a detailed discussion on other options, see also Macháček and Bojar (2014).

Whether or not a given comparison of a pair of distinct translations of the same source input, s_1 and s_2, is counted as a concordant (Conc) or disconcordant (Disc) pair is defined by the following matrix:

In previous years, we used bootstrap resampling (Koehn, 2004; Graham et al., 2014) to estimate confidence intervals for our Kendall's Tau formulation, and metrics with non-overlapping 95% confidence

	en-de 11	en-de$_B$ 11	en-de$_P$ 11	en-de$_M$ 11	en-zh 12	en-zh$_B$ 12	en-zh$_M$ 12	de-en 9	de-en$_B$ 9	de-en$_M$ 9	ru-en 10	ru-en$_B$ 10	ru-en$_M$ 10	zh-en 15	zh-en$_B$ 15	zh-en$_M$ 15
SENTBLEU	0.823	0.837	0.815	**0.827**	0.927	0.911	0.919	**0.786**	0.763	**0.788**	**0.833**	**0.850**	**0.837**	**0.950**	0.928	**0.944**
BLEU	0.825	0.844	0.830	0.822	0.928	0.899	0.913	0.778	0.797	**0.805**	0.761	0.780	0.775	**0.957**	**0.934**	**0.949**
TER	**0.848**	**0.860**	**0.859**	**0.852**	-0.213	-0.200	-0.203	**0.766**	**0.744**	**0.758**	**0.829**	**0.832**	**0.853**	0.911	0.875	0.911
CHRF++	**0.850**	**0.866**	**0.876**	**0.858**	0.878	0.915	0.885	0.699	0.681	0.704	**0.833**	**0.839**	**0.843**	**0.955**	**0.948**	**0.952**
CHRF	**0.862**	**0.874**	**0.883**	–	0.923	0.912	–	0.687	0.683	–	0.831	**0.839**	–	**0.954**	**0.947**	–
PARBLEU	0.774	0.796	0.724	0.794	0.962	0.955	0.959	**0.838**	**0.831**	**0.829**	0.744	0.767	0.756	**0.953**	**0.934**	**0.945**
PARCHRF++	**0.845**	**0.863**	**0.865**	**0.856**	0.948	**0.966**	0.896	0.708	0.704	0.669	0.823	**0.834**	**0.832**	**0.956**	**0.950**	**0.956**
CHARACTER	**0.868**	**0.889**	**0.835**	**0.878**	0.905	0.908	0.901	0.687	0.696	**0.713**	**0.869**	**0.853**	**0.873**	**0.950**	**0.942**	**0.949**
EED	**0.869**	**0.871**	**0.867**	**0.867**	0.928	0.923	0.930	**0.752**	**0.747**	**0.752**	**0.872**	**0.868**	**0.879**	0.932	0.922	0.932
YISI-0	**0.889**	**0.882**	**0.873**	**0.886**	0.362	0.273	0.332	**0.786**	**0.790**	**0.794**	**0.874**	**0.867**	**0.880**	0.918	0.911	0.918
SWSS+METEOR	–	–	–	–	–	–	–	–	–	–	**0.876**	**0.891**	**0.891**	0.926	0.923	0.929
MEE	0.820	0.833	0.830	**0.820**	–	–	–	0.712	0.674	0.712	**0.878**	**0.876**	**0.876**	**0.948**	**0.940**	**0.948**
PRISM	**0.851**	**0.854**	**0.839**	**0.851**	0.221	0.178	0.221	**0.775**	**0.763**	**0.770**	**0.839**	**0.841**	**0.842**	**0.945**	**0.949**	**0.945**
YISI-1	**0.887**	**0.886**	**0.888**	**0.885**	0.959	0.959	0.960	**0.783**	**0.781**	**0.781**	**0.833**	**0.837**	**0.838**	**0.942**	**0.942**	**0.943**
YISI-COMBI	0.868	0.873	0.876	**0.876**	–	–	–	–	–	–	–	–	–	–	–	–
BLEURT-YISI-COMBI	0.868	0.873	0.876	**0.876**	–	–	–	–	–	–	–	–	–	–	–	–
BERT-BASE-L2	–	–	–	–	–	–	–	0.791	**0.798**	**0.802**	**0.836**	**0.833**	**0.835**	0.929	**0.936**	**0.933**
BERT-LARGE-L2	–	–	–	–	–	–	–	0.800	**0.801**	**0.812**	**0.843**	**0.844**	**0.850**	0.928	**0.935**	**0.932**
MBERT-L2	**0.861**	**0.862**	0.841	**0.865**	0.934	0.925	0.936	**0.824**	**0.825**	**0.834**	0.805	0.813	0.816	**0.935**	**0.938**	**0.939**
BLEURT	–	–	–	–	–	–	–	0.770	**0.769**	0.780	**0.844**	**0.847**	**0.850**	**0.931**	**0.936**	**0.935**
BLEURT-EXTENDED	**0.870**	**0.870**	**0.860**	**0.867**	0.928	0.923	0.925	**0.818**	**0.805**	**0.812**	0.797	0.793	0.795	0.931	0.932	0.932
ESIM	**0.894**	**0.900**	**0.887**	**0.898**	0.972	**0.975**	0.976	0.808	**0.798**	**0.804**	**0.834**	**0.842**	**0.839**	0.910	**0.920**	0.916
PARESIM-1	**0.886**	**0.897**	**0.878**	**0.890**	**0.983**	**0.983**	**0.985**	**0.835**	**0.807**	**0.822**	**0.828**	**0.840**	**0.835**	0.910	0.918	0.915
COMET	**0.863**	**0.864**	**0.858**	**0.864**	0.007	-0.014	-0.004	0.773	**0.769**	0.772	**0.836**	**0.836**	**0.836**	**0.931**	**0.936**	**0.934**
COMET-2R	**0.869**	**0.869**	**0.866**	**0.875**	-0.066	-0.076	-0.075	0.772	**0.764**	0.771	**0.843**	**0.842**	**0.843**	**0.928**	**0.930**	**0.929**
COMET-HTER	**0.852**	**0.855**	**0.848**	**0.853**	-0.073	-0.075	-0.074	**0.767**	**0.769**	**0.768**	0.741	0.744	0.742	0.873	0.869	0.871
COMET-MQM	0.840	0.844	0.836	0.842	0.084	0.076	0.080	**0.684**	**0.686**	**0.685**	0.746	0.750	0.748	0.862	0.860	0.861
COMET-RANK	**0.860**	0.839	**0.831**	**0.852**	0.540	0.507	0.530	**0.757**	0.582	**0.723**	0.732	0.743	0.757	**0.909**	**0.908**	**0.919**

Table 8: Influence of references: Pearson correlation of metrics with DA human assessment for MT systems excluding outliers in WMT2020 for all language-pairs with multiple references; correlations of metrics not significantly outperformed by any other for that language pair are highlighted in bold. The subscript *B* represents a secondary reference, *P* represents a paraphrased reference, *M* represents all available references.
Note that we exclude reference-free metrics from this table, so the winners are not comparable with the main tables.

		Metric		
		$s_1 < s_2$	$s_1 = s_2$	$s_1 > s_2$
Human	$s_1 < s_2$	Conc	Disc	Disc
	$s_1 = s_2$	–	–	–
	$s_1 > s_2$	Disc	Disc	Conc

intervals are identified as having statistically significant difference in performance. The tests are inconclusive for most metric pairs this year and we do not include them in the paper.

4.2.1 Segment-Level Results

Results of the segment-level human evaluation for translations sampled from the News Translation Task are shown in Tables 9 and 10, We expect that comparing between segments translated by two MT systems that are far apart in quality would be a relatively easier task for automatic metrics. So we also include results after discarding segments that were translated by outlier systems.

Note that we do not include any human-translated segments in this evaluation.

4.3 Document-level Results

Results of the document-level human evaluation for translations sampled from the News Translation Task are shown in Tables 11 and 12.

5 Discussion

5.1 System-Level Results

In general, there is no clear best metric this year across all language pairs. For most language pairs, the William's significance test results in large clusters of metrics. The set of "winners" according to the test (i.e., the metrics that are not outperformed by any other metric) are typically not consistent across language pairs.

The sample of systems we employ to evaluate metrics is often small, as few as six MT systems for Pashto → English, for example. This can lead to inconclusive results, as identification of significant differences in correlations of metrics is unlikely at such a small sample size. Furthermore, Williams test takes into account the correlation between each pair of metrics, in addition to the correlation between the metric scores themselves, and this latter correlation increases the likelihood of a significant difference being identified. In extreme cases, the test would have low power when comparing a metric that doesn't correlate well with other metrics, resulting in this metric not being outperformed by other metrics despite having a much lower value of correlation.

To strengthen the conclusions of our evaluation, in past years (Bojar et al., 2016, 2017; Ma et al., 2018), we included significance test results for large hybrid-super-samples of systems 10K hybrid systems were created per language pair, with corresponding DA human assessment scores by sampling pairs of systems from the News Translation Task, creating hybrid systems by randomly selecting each candidate translation from one of the two selected systems. However, as WMT human annotations are collected with document context in 2020, this style of hybridization is susceptible to breaking cross-segment references in MT outputs and it would be unreasonable to shuffle individual segments. The creation of hybrid systems would need to be done by sampling documents instead of segments from all sets of systems. Finally, it is possible that including documents translated by outlier systems might falsely lead to high correlations. We believe that this merits further investigation based on data from previous of metrics tasks, and we do not attempt it this year.

In the rest of this section, we present analysis of various aspects of system-level evaluation based on scatterplots of all metrics. Appendix B contains scatterplots of metrics for each language pair. We include BLEU, chrF, the "best" reference-based metric and the "best" reference-free metric (we acknowledge that this is not the best way to define the best metric, but we choose the metric that is most highly correlated with humans on the set of all MT systems after removing outliers).

5.1.1 Influence of Domain in English → Inuktitut

English → Inuktitut training data was the Canadian Hansards domain, and the development data contained a small amount of news data. The test set was a mix of in-domain data from the Hansards and news documents. The evaluation was only done on the out-of-domain news documents, so we also look at metric scores computed only on the subset of news sentences.

Figure 1 shows that BLEU scores on the out-of-domain dataset are considerably smaller than the full dataset, showing that MT systems have a higher quality on the in-domain dataset. The relative scores of metrics remain mostly stable when we compare scores on the full test set to scores on

	cs-en		de-en		iu-en		ja-en		km-en	pl-en		ps-en	ru-en		ta-en		zh-en	
	all	all-out	all	all-out	all	all-out	all	all-out	all	all	all-out	all	all	all-out	all	all-out	all	all-out
	14018	9461	16584	6185	8162	5381	15193	6286	3706	21121	17979	3507	14024	11020	12789	8749	62586	53610
SENTBLEU	0.068	0.057	0.413	-0.025	0.182	0.170	0.188	0.061	0.226	-0.024	-0.046	0.096	-0.005	-0.038	0.162	0.069	0.093	0.060
TER	-0.04	-0.06	0.355	-0.137	0.021	0.012	0.044	-0.077	0.125	-0.172	-0.196	-0.036	-0.117	-0.154	0.046	-0.063	-0.01	-0.047
CHRF++	0.090	0.075	0.435	0.013	0.246	0.251	0.245	0.115	0.275	0.034	0.009	0.145	0.054	0.018	0.186	0.098	0.130	0.096
CHRF	0.086	0.072	0.438	0.018	0.254	0.260	0.242	0.109	0.267	0.028	0.003	0.144	0.049	0.012	0.186	0.096	0.132	0.098
PARBLEU	0.058	0.038	0.415	-0.039	0.167	0.161	0.198	0.074	0.203	-0.025	-0.049	0.100	-0.011	-0.052	0.159	0.064	0.095	0.059
PARCHRF++	0.096	0.082	0.436	0.009	0.232	0.235	0.247	0.117	0.267	0.027	0.002	0.147	0.044	0.007	0.184	0.095	0.132	0.099
CHARACTER	0.090	0.087	0.440	0.011	0.214	0.220	0.221	0.106	0.248	0.023	-0.002	0.172	0.057	0.028	0.138	0.078	0.123	0.093
EED	0.091	0.078	0.440	0.013	0.256	0.258	0.235	0.116	0.271	0.045	0.022	0.149	0.053	0.018	0.198	0.103	0.129	0.093
YISI-0	0.072	0.065	0.441	0.024	0.261	0.263	0.241	0.121	0.268	0.035	0.013	0.140	0.065	0.030	0.183	0.089	0.127	0.093
SWSS+METEOR	–		–		0.226	0.218	0.228	0.086	0.264	0.011	-0.016	0.130	0.048	0.010	0.205	0.120	0.133	0.099
MEE	0.063	0.045	0.402	-0.04	0.134	0.126	0.187	0.064	0.206	-0.084	-0.105	0.078	-0.041	-0.084	0.114	0.032	0.083	0.050
YISI-1	0.117	0.103	0.468	0.051	0.253	0.260	0.277	0.128	0.316	0.042	0.023	0.147	0.091	0.049	0.248	0.162	0.146	0.115
BERT-BASE-L2	0.103	0.087	0.454	0.026	0.238	0.229	0.263	0.129	0.295	0.032	0.013	0.159	0.087	0.037	0.223	0.135	0.141	0.113
BERT-LARGE-L2	0.102	0.087	0.456	0.025	0.251	0.249	0.262	0.114	0.314	0.044	0.027	0.151	0.094	0.047	0.245	0.157	0.133	0.102
MBERT-L2	0.119	0.111	0.442	0.001	0.244	0.235	0.251	0.120	0.312	0.047	0.029	0.151	0.083	0.036	0.227	0.139	0.133	0.104
BLEURT	0.126	0.118	0.456	0.015	0.258	0.256	0.265	0.123	0.327	0.057	0.040	0.207	0.093	0.046	0.230	0.145	0.137	0.107
BLEURT-EXTENDED	0.127	0.113	0.448	0.004	0.259	0.259	0.271	0.124	0.330	0.044	0.019	0.161	0.101	0.057	0.246	0.165	0.137	0.107
ESIM	0.110	0.103	0.454	0.031	0.241	0.233	0.239	0.119	0.300	0.058	0.045	0.147	0.084	0.044	0.208	0.117	0.138	0.108
PARESIM-1	0.105	0.098	0.464	0.051	0.249	0.241	0.242	0.121	0.292	0.066	0.055	0.149	0.089	0.049	0.213	0.123	0.139	0.111
COMET	0.129	0.112	0.485	0.090	0.281	0.271	0.274	0.127	0.298	0.099	0.085	0.158	0.156	0.117	0.241	0.163	0.171	0.142
COMET-2R	0.120	0.107	0.479	0.101	0.257	0.251	0.268	0.120	0.308	0.098	0.085	0.144	0.148	0.110	0.253	0.177	0.163	0.136
COMET-HTER	0.103	0.087	0.481	0.088	0.198	0.199	0.241	0.095	0.269	0.080	0.067	0.116	0.131	0.098	0.227	0.151	0.135	0.113
COMET-MQM	0.108	0.097	0.483	0.100	0.215	0.209	0.259	0.112	0.282	0.080	0.066	0.141	0.137	0.102	0.227	0.158	0.141	0.117
COMET-RANK	0.099	0.096	0.470	0.061	0.188	0.181	0.235	0.086	0.228	0.073	0.057	0.107	0.118	0.082	0.199	0.112	0.142	0.117
COMET-QE	0.091	0.072	0.410	0.042	0.031	0.020	0.153	0.048	0.148	0.039	0.029	0.092	0.084	0.049	0.163	0.099	0.088	0.070
OPENKIWI-BERT	0.036	0.029	0.379	0.013	-0.005	-0.009	0.110	0.000	0.168	-0.033	-0.043	0.076	-0.033	-0.067	0.118	0.052	0.029	0.020
OPENKIWI-XLMR	0.093	0.079	0.463	0.074	0.056	0.031	0.220	0.086	0.244	0.059	0.051	0.106	0.092	0.065	0.188	0.109	0.115	0.089
YISI-2	0.068	0.054	0.413	0.006	0.039	0.028	0.204	0.074	0.214	0.048	0.042	0.073	0.070	0.056	0.199	0.113	0.116	0.084
PRISM	0.143	0.135	0.475	0.057	0.255	0.254	0.272	0.146	0.304	0.109	0.093	0.165	0.145	0.111	0.237	0.151	0.167	0.138
BAQ_DYN	–		–		–		–		–	–		–	–		–		0.119	0.089
BAQ_STATIC	–		–		–		–		–	–		–	–		–		0.119	0.087

Table 9: Segment-level metric results for to-English language pairs: Kendall's Tau formulation of segment-level metric scores with DARR scores. For language pairs that contain outlier systems, we also show correlation after discarding segments translated by outlier systems

	en-cs		en-de		en-iu		en-ja	en-pl		en-ru	en-ta		en-zh
	all	all-out	all	all-out	all	all-out	all	all	all-out	all	all	all-out	all
	21121	10283	9339	4637	13159	5490	12830	17689	9316	8330	9087	3695	12652
SENTBLEU	0.432	0.194	0.303	0.155	0.206	-0.084	0.479	0.153	0.067	0.051	0.398	0.206	0.396
TER	0.317	0.067	0.182	0.044	-0.071	-0.337	-0.591	0.003	-0.094	-0.121	0.203	0.019	-0.36
CHRF++	0.478	0.228	0.367	0.215	0.338	0.075	0.506	0.255	0.154	0.156	0.579	0.349	0.388
CHRF	0.472	0.229	0.379	0.224	0.344	0.095	0.506	0.250	0.150	0.153	0.589	0.359	0.400
PARBLEU	0.460	0.226	0.299	0.136	0.212	-0.051	0.052	0.183	0.088	0.062	0.340	0.178	0.356
PARCHRF++	0.492	0.253	0.355	0.192	–		0.527	0.272	0.167	0.176	–		0.398
CHARACTER	0.413	0.195	0.311	0.179	0.309	0.108	0.471	0.198	0.107	0.143	0.525	0.270	0.339
EED	0.458	0.210	0.363	0.203	0.361	0.109	0.515	0.248	0.151	0.155	0.587	0.342	0.393
YISI-0	0.432	0.191	0.349	0.212	0.362	0.101	0.484	0.233	0.132	0.151	0.547	0.336	0.319
MEE	0.411	0.157	0.289	0.128	-0.074	-0.272	–	0.125	0.025	0.027	0.373	0.168	–
YISI-1	0.550	0.320	0.427	0.263	0.251	0.082	0.568	0.349	0.209	0.256	0.669	0.440	0.463
YISI-COMBI	–		0.399	0.224	–		–	–		–	–		–
BLEURT-COMBI	–		0.399	0.224	–		–	–		–	–		–
MBERT-L2	0.567	0.359	0.361	0.202	–		0.541	0.350	0.212	0.246	0.587	0.334	0.432
BLEURT-EXTENDED	0.689	0.517	0.447	0.278	0.359	0.112	0.533	0.430	0.271	0.305	0.643	0.419	0.460
ESIM	0.469	0.253	0.347	0.195	0.122	-0.018	0.522	0.312	0.203	0.224	0.599	0.363	0.391
PARESIM-1	0.475	0.257	0.343	0.197	0.122	-0.018	0.510	0.324	0.209	0.230	0.599	0.363	0.396
COMET	0.668	0.487	0.468	0.324	0.322	0.078	0.624	0.462	0.316	0.344	0.671	0.457	0.432
COMET-2R	0.669	0.512	0.463	0.321	0.326	0.078	0.630	0.445	0.294	0.343	0.676	0.463	0.434
COMET-HTER	0.665	0.500	0.440	0.303	0.331	0.088	0.601	0.427	0.274	0.292	0.640	0.411	0.411
COMET-MQM	0.666	0.490	0.423	0.275	0.313	0.078	0.588	0.424	0.271	0.281	0.635	0.413	0.388
COMET-RANK	0.629	0.408	0.379	0.217	0.297	0.097	0.569	0.388	0.207	0.229	0.588	0.342	0.380
COMET-QE	0.614	0.470	0.347	0.233	-0.04	-0.051	0.470	0.360	0.211	0.264	0.514	0.320	0.346
OPENKIWI-BERT	0.262	0.142	0.168	0.058	-0.115	-0.233	-0.529	0.153	0.035	0.164	0.169	0.022	0.077
OPENKIWI-XLMR	0.607	0.417	0.369	0.224	0.060	0.009	0.553	0.347	0.189	0.279	0.604	0.354	0.377
YISI-2	0.187	0.104	0.296	0.171	0.146	0.073	0.383	0.115	0.052	0.146	0.545	0.332	0.152
PRISM	0.619	0.414	0.447	0.280	0.452	0.195	0.579	0.414	0.274	0.283	0.448	0.211	0.397
BAQ_DYN	–		–		–		–	–		–	–		0.351
BAQ_STATIC	–		–		–		–	–		–	–		0.344
EQ_DYN	–		–		–		–	–		–	–		0.356
EQ_STATIC	–		–		–		–	–		–	–		0.409

Table 10: Segment-level metric results for out-of-English language pairs: Kendall's Tau formulation of segment-level metric scores with DARR scores; For language pairs that contain outlier systems, we also show correlation after discarding segments translated by outlier systems

	cs-en		de-en		iu-en		ja-en		pl-en		ru-en		ta-en		zh-en	
	all	all-out	all	all-out	all	all-out	all	all-out	all	all-out	all	all-out	all	all-out	all	all-out
	1424	955	1866	495	36	24	790	311	635	529	753	581	684	440	3085	2618
SENTBLEU	0.104	0.058	0.601	-0.055	0.611	0.417	0.413	0.125	0.096	0.059	0.113	0.026	0.330	0.150	0.211	0.153
TER	0.115	0.068	0.621	-0.002	0.611	0.500	0.370	0.048	0.024	-0.059	0.089	-0.009	0.383	0.214	0.197	0.141
CHRF++	0.135	0.110	0.624	0.006	0.500	0.333	0.435	0.158	0.071	0.036	0.169	0.088	0.395	0.209	0.199	0.139
CHRF	0.126	0.091	0.626	0.002	0.611	0.417	0.453	0.209	0.065	0.021	0.195	0.112	0.395	0.200	0.209	0.154
PARBLEU	0.100	0.045	0.630	-0.002	0.556	0.417	0.428	0.138	0.065	0.032	0.086	-0.019	0.368	0.177	0.201	0.143
PARCHRF++	0.117	0.081	0.642	0.042	0.611	0.417	0.438	0.164	0.087	0.040	0.171	0.095	0.412	0.232	0.203	0.146
CHARACTER	0.059	0.049	0.646	0.079	0.500	0.250	0.410	0.145	0.090	0.051	0.187	0.122	0.371	0.196	0.219	0.166
EED	0.105	0.064	0.633	0.006	0.722	0.583	0.430	0.125	0.080	0.017	0.174	0.088	0.395	0.200	0.206	0.148
YISI-0	0.052	0.022	0.616	0.006	0.556	0.333	0.425	0.125	0.071	0.036	0.187	0.098	0.409	0.223	0.196	0.139
SWSS+METEOR	–		–		0.722	0.583	0.377	0.029	0.109	0.047	0.211	0.129	0.447	0.291	0.201	0.141
MEE	0.126	0.114	0.618	-0.006	0.444	0.250	0.438	0.190	0.014	-0.013	0.137	0.053	0.398	0.245	0.198	0.140
YISI-1	0.136	0.114	0.640	0.034	0.667	0.500	0.420	0.119	0.109	0.062	0.150	0.033	0.450	0.300	0.210	0.153
BERT-BASE-L2	0.164	0.139	0.654	0.075	0.778	0.667	0.430	0.151	0.046	-0.013	0.179	0.064	0.398	0.223	0.206	0.149
BERT-LARGE-L2	0.131	0.091	0.642	0.030	0.722	0.583	0.418	0.119	0.027	-0.028	0.195	0.084	0.439	0.291	0.185	0.124
MBERT-L2	0.149	0.118	0.621	-0.006	0.833	0.750	0.433	0.158	0.033	-0.036	0.232	0.126	0.418	0.259	0.216	0.162
BLEURT	0.154	0.125	0.641	0.038	0.667	0.500	0.420	0.100	0.039	-0.009	0.227	0.115	0.418	0.259	0.197	0.141
BLEURT-EXTENDED	0.140	0.114	0.633	0.014	0.833	0.750	0.430	0.113	0.077	0.006	0.243	0.143	0.412	0.245	0.198	0.141
ESIM	0.135	0.110	0.670	0.164	0.722	0.583	0.400	0.087	0.039	-0.017	0.174	0.064	0.404	0.236	0.203	0.148
PARESIM-1	0.119	0.093	0.670	0.156	0.722	0.583	0.392	0.055	0.033	-0.021	0.171	0.060	0.401	0.232	0.208	0.154
COMET	0.142	0.114	0.626	-0.018	0.667	0.500	0.392	0.061	0.112	0.070	0.193	0.088	0.395	0.218	0.206	0.151
COMET-2R	0.138	0.116	0.614	0.030	0.778	0.667	0.413	0.093	0.090	0.047	0.227	0.136	0.404	0.232	0.214	0.158
COMET-HTER	0.160	0.133	0.638	0.042	0.556	0.333	0.415	0.138	0.083	0.040	0.169	0.084	0.354	0.191	0.150	0.105
COMET-MQM	0.140	0.114	0.645	0.075	0.611	0.417	0.410	0.119	0.080	0.043	0.163	0.081	0.386	0.241	0.161	0.117
COMET-RANK	0.139	0.131	0.615	-0.026	0.667	0.500	0.365	0.035	0.112	0.096	0.185	0.074	0.325	0.154	0.199	0.147
COMET-QE	0.091	0.060	0.636	0.042	0.389	0.250	0.329	0.023	-0.002	-0.028	0.153	0.060	0.301	0.127	0.169	0.118
OPENKIWI-BERT	0.087	0.064	0.628	0.046	0.444	0.250	0.322	0.113	0.096	0.077	0.137	0.050	0.281	0.145	0.113	0.079
OPENKIWI-XLMR	0.133	0.114	0.613	0.010	0.556	0.500	0.418	0.145	0.055	0.017	0.155	0.060	0.389	0.227	0.187	0.135
YISI-2	0.083	0.072	0.547	-0.075	0.278	0.250	0.385	0.055	0.118	0.153	0.248	0.195	0.383	0.196	0.199	0.139
PRISM	0.169	0.156	0.636	-0.002	0.667	0.500	0.420	0.119	0.102	0.059	0.211	0.102	0.406	0.236	0.195	0.138
BAQ_DYN	–		–		–		–		–		–		–		0.223	0.172
BAQ_STATIC	–		–		–		–		–		–		–		0.214	0.160

Table 11: Document-level metric results for to-English language pairs: Kendall's Tau formulation of segment-level metric scores with DAdocument-level metric scores with DOC-DARR judgements. For language pairs that contain outlier systems, we also show correlation after discarding documents translated by outlier systems.

	en-cs		en-de		en-iu		en-ja	en-pl		en-ru	en-ta		en-zh
	all	all-out	all	all-out	all	all-out	all	all	all-out	all	all	all-out	all
	1442	572	729	312	203	48	469	677	254	387	389	99	651
SENTBLEU	0.680	0.273	0.550	0.359	0.596	-0.25	0.808	0.510	0.150	0.287	0.799	0.596	0.598
TER	0.691	0.294	0.517	0.308	0.567	-0.292	-0.07	0.439	0.094	0.178	0.748	0.616	0.118
CHRF++	0.692	0.294	0.583	0.372	0.547	-0.417	0.829	0.536	0.165	0.339	0.866	0.596	0.579
CHRF	0.688	0.290	0.597	0.397	0.576	-0.333	0.838	0.524	0.165	0.307	0.872	0.616	0.591
PARBLEU	0.727	0.381	0.528	0.269	0.576	-0.208	0.565	0.569	0.236	0.266	0.805	0.596	0.625
PARCHRF++	0.717	0.364	0.575	0.340	–		0.825	0.560	0.205	0.307	–		0.650
CHARACTER	0.656	0.224	0.520	0.263	0.547	-0.292	0.842	0.448	0.047	0.328	0.872	0.657	0.613
EED	0.678	0.280	0.569	0.340	0.596	-0.292	0.834	0.554	0.228	0.277	0.856	0.556	0.588
YISI-0	0.653	0.245	0.553	0.327	0.645	-0.125	0.821	0.554	0.228	0.318	0.830	0.515	0.441
MEE	0.714	0.357	0.558	0.314	0.527	-0.375	–	0.489	0.071	0.307	0.805	0.495	–
YISI-1	0.763	0.462	0.605	0.359	0.616	-0.083	0.855	0.663	0.339	0.349	0.882	0.657	0.690
YISI-COMBI	–		0.594	0.353	–		–	–		–	–		–
BLEURT-COMBI	–		0.594	0.353	–		–	–		–	–		–
MBERT-L2	0.781	0.517	0.580	0.308	–		0.842	0.648	0.299	0.431	0.861	0.596	0.693
BLEURT-EXTENDED	0.847	0.664	0.635	0.378	0.635	-0.167	0.851	0.740	0.488	0.437	0.856	0.636	0.708
ESIM	0.720	0.392	0.534	0.237	0.507	-0.25	0.855	0.616	0.315	0.354	0.836	0.596	0.674
PARESIM-1	0.741	0.441	0.528	0.237	0.507	-0.25	0.829	0.628	0.331	0.364	0.836	0.596	0.662
COMET	0.845	0.664	0.632	0.404	0.547	-0.125	0.847	0.687	0.346	0.359	0.897	0.758	0.561
COMET-2R	0.859	0.699	0.613	0.391	0.596	-0.25	0.868	0.725	0.409	0.375	0.866	0.636	0.558
COMET-HTER	0.849	0.675	0.616	0.410	0.606	-0.042	0.855	0.669	0.307	0.287	0.856	0.657	0.502
COMET-MQM	0.849	0.675	0.594	0.372	0.567	-0.125	0.817	0.678	0.291	0.364	0.836	0.657	0.472
COMET-RANK	0.803	0.573	0.572	0.321	0.586	-0.375	0.812	0.607	0.165	0.307	0.841	0.596	0.472
COMET-QE	0.839	0.650	0.514	0.250	0.212	-0.167	0.812	0.619	0.142	0.297	0.820	0.657	0.469
OPENKIWI-BERT	0.655	0.399	0.443	0.147	0.488	0.083	0.139	0.427	0.024	0.344	0.584	0.111	0.459
OPENKIWI-XLMR	0.821	0.622	0.589	0.314	0.527	0.167	0.859	0.592	0.094	0.328	0.856	0.596	0.524
YISI-2	0.255	0.105	0.416	0.224	0.527	0.333	0.680	0.117	-0.118	0.209	0.805	0.434	0.223
PRISM	0.792	0.545	0.594	0.378	0.665	-0.042	0.829	0.634	0.283	0.328	0.733	0.374	0.511
BAQ_DYN	–		–		–		–	–		–	–		0.567
BAQ_STATIC	–		–		–		–	–		–	–		0.619
EQ_DYN	–		–		–		–	–		–	–		0.613
EQ_STATIC	–		–		–		–	–		–	–		0.644

Table 12: Document-level metric results for out-of-English language pairs: Kendall's Tau formulation of document-level metric scores with DOC-DARR judgements. For language pairs that contain outlier systems, we also show correlation after discarding documents translated by outlier systems

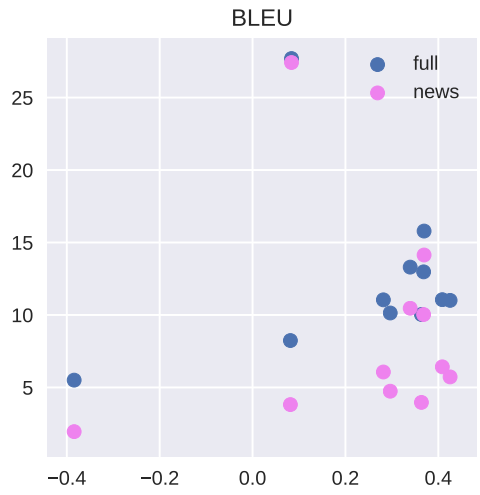


Figure 1: English → Inuktitut: Human vs. BLEU scores on the full dataset vs. the news subset. Only the news subset was included in the human evaluation. Each dot corresponds to an MT system, the outlier on the top-right is UQAM_TANLE.

only the news subset that was evaluated. The main exception is UQAM_TANLE; BLEU scores are really high on the out-of-domain data, and increase very little when computed on the full dataset. When looking at correlations with human scores (Table 7), we expected correlations to increase when computed over the news subset. This is true for most metrics such as COMET-QE, but the correlation stays the same or actually decreases for other metrics like PARBLEU.

5.1.2 Scoring Human Translations

The alternate reference was included in the manual evaluation for German → English, Russian → English and Chinese → English. All human references were included in the out-of-English manual evaluation.[7]

German → English: HUMAN-B was ranked third in the manual evaluation. The lexical metrics (BLEU, CHRF, CHARACTER, EED, MEE, YISI-0, CHRF++, PARBLEU, PARCHRF) give extremely low scores to the HUMAN-B reference. This is also true for PRISM and all the reference-free metrics except COMET-QE. The neural metrics also give low scores to the human reference, however, the margin of error is much smaller. COMET-QE is the only metric that gives high scores to the HUMAN-B reference.

Appendix B also shows the scatterplot "newstestB2020" where HUMAN-B served as the reference for the metrics. We see some differences in the vertical axis but the general picture remains the same even with this fairly different human translation.

Russian → English: The HUMAN-B reference was ranked after 6 MT systems in the manual evaluation but still within the same cluster, so not significantly distinguishable. Lexical metrics give relatively low scores to HUMAN-B. The neural metrics give relatively higher scores, but score it above *Online-A* and below *ariel197197*, i.e. differently than DA judgements.

Chinese → English: The Human translation is ranked 12th in the manual evaluation (in a giant cluster which puts together all but one top and one bottom system), and most metrics place it more or less correctly. Many metrics, including lexical metrics, still have correlations above 0.9 even after including the Human translation.

English → German: According to the WMT human evaluation, the HUMAN-B reference receives the highest scores, the HUMAN-A reference is ranked fourth and Human-P, which was generated by linguists paraphrasing the WMT references, is ranked lower at 10th place. Each human reference falls into a separate cluster of significance.

Lexical metrics score around 10 MT systems above each WMT reference (using the other WMT human translation as reference). COMET-QE and some neural metrics (BLEURT, COMET-MQM, COMET-HTER and MBERT-L2) score HUMAN-A and HUMAN-B as better than all MT systems.

When using either of the WMT references, most metrics, including all the lexical metrics, score the paraphrased reference much lower than the rest of the systems. The COMET family of metrics and BLEURT-EXTENDED are the only metrics that are able to recognise the merit of the paraphrased references.

When using the paraphrased references, all reference-based metrics score the two human translations above all MT systems, often by a large margin. PRISM is the sole exception; it scores the HUMAN-B reference about half way between the MT systems. Interestingly, most of these metrics score HUMAN-A above HUMAN-B, i.e. dis-

[7] Findings 2020 in the official tables label the alternate reference in into-English direction simply as HUMAN. The "first" reference, which serves as the primary reference for us, was not scored manually in DA into English. Out of English, the primary reference for us is labelled HUMAN-A in Findings.

agreeing with DA judgements. Metric correlations when including HUMAN-A system drop dramatically when using the alternate WMT reference, but the correlations are higher with the paraphrased reference. This also holds when scoring HUMAN-B using the paraphrased vs the main WMT reference (Table 7).

Of the reference-free metrics, COMET-QE scores the two WMT references above all MT systems, and ranks the paraphrased reference similar to its rank in the manual evaluation. OPENKIWI-BERT and OPENKIWI-XLMR are a little biased against these human translations, and YISI-2 scores all human translations below all MT systems.

English → Chinese The manual evaluation ranks the two Human translations above all MT systems, but most metrics give these much low scores.

To summarize, we see that the current MT metrics generally struggle to score human translations against machine translations reliably. Rare exceptions include primarily trained neural metrics and reference-less COMET-QE. While the metrics are not really prepared to score human translations, we find this type of test relevant as more and more language pairs are getting closer to the human translation benchmark. A general-enough metric should be thus able to score human translation comparably and not rely on some idiosyncratic properties of MT outputs. We hope that human translations will be included in WMT DA scoring in the upcoming years, too.

5.1.3 Influence of Outliers

There are no outlier systems for some language-pairs like Khmer→ English and English → Russian. For others, we have systems whose score is far away from the scores of the rest of the systems. As these outliers have a large influence on Pearson correlation, computing the correlation without outliers typically makes the task harder for metrics and results in a decrease in correlation.

For example, we identify three outliers in the German → English set; the quality of the last system is extremely low compared to rest. All reference-based metrics have high correlations when including all systems, but correlations drop when discarding outliers. In particular, CHRF and PARESIM both had a correlation of 0.95 when computed over all systems, but this drops to 0.69 and 0.83 respectively after removing outliers, revealing that PARESIM is more reliable with this language pair. An even larger drop is observed for CHRF and CHRF++ in English → Czech, from 0.8 to 0.3. We find this particularly surprising because CHRF has always performed well on this language pair, including in the evaluation on the gradually reducing set of top N systems, i.e. in harder and harder conditions, see SACREBLEU-CHRF in Appendix A.4 of Ma et al. (2019).

In some cases, metrics can be inaccurate when scoring outliers, resulting in an increased correlation when correlation is recomputed over non-outlier systems. For example, with Chinese → English, the score of WMTBIOMEDBASELINE score is much lower than the next system. Most metrics correctly rank it last as well, but COMET-HTER, COMET-MQM, COMET-QE and OPENKIWI-BERT give it a higher score than the next system(s). Note that the other metrics all have a correlation of above 0.9 even after removing the outlier.

In other cases, removing outliers decreases the correlation of a metric and yet it helps its final outcome. For instance SENTBLEU averaged over all sentences becomes one of the "winners" in the system-level evaluation of translation into English (Table 5). If we trust the results without outliers more, using *averaged sentBLEU* seems better than using plain old BLEU and not significantly worse than any other metric going from English into several target languages.

For some language-pairs, we override the decisions made by the outlier detection algorithm, based on whether we believe including or removing these systems from consideration would have an impact on the correlations: For example, with Tamil → English, the last two systems are not classified as outliers by the algorithm, but their human scores is some distance away from the rest of the systems. CHRF, CHRF++ and PARCHRF++ are the only metrics that correctly order these two systems. OPENKIWI-BERT and OPENKIWI-XLMR both get these two systems wrong with a large margin. But for all metrics, removing these systems leads to a significant drop in correlation. Thus we count these two systems as outliers.

Another example is Japanese → English. For this language-pair, we have two clusters of 7 and 3 systems. Metrics have high correlations when considering all systems, but when looking at MT

systems within individual clusters, there are discrepancies between the metric scores compared to human scores. The outlier detection algorithm flags only the last two systems as outliers, but the presence of the third system has a disproportionate impact on the correlation. We include all three systems in the set of outliers.

The influence of references For all language pairs where multiple references were available, the correlations are typically very close whether using the primary reference or the alternate reference. For metrics where we do see a difference, there is no consistent pattern whether metrics prefer one reference or the other. We note that although the change in correlations is small when comparing across reference sets, the set of "winners" according to the William's test for statistical significance is not stable, particularly for English → German. When combining references, in most cases, the correlation with multiple references lies between the correlation of the individual references. For example, with English → German, BLEU correlates best with the secondary reference with a correlation of 0.844. But with multiple references, the correlation is 0.825, just above the correlation with the primary reference with is 0.822 (Table 8).

There are a few exceptions where there is a small increase in metric correlation above both individual references. For example, the correlation of CHARACTER with German → English increases from 0.687 and 0.696 with a single reference to 0.713 with both references (Table 8). But there are no metrics which consistently show an improvement with multiple references across multiple language pairs.

5.1.4 Neural vs. Lexical Metrics

For many language pairs, when we look at correlation clustering of the reference-based metrics based on their system-level scores, we end up with two major clusters: neural metrics and lexical metrics. We have seen that lexical and neural metrics differ in how they score the human translations. For English → German, all lexical metrics have a slightly higher correlation than any neural metric when evaluating MT systems. However, these metrics make major errors evaluating the HUMAN-A translations with the HUMAN-B reference.

We also see such differences with some MT systems. Selected examples:

- English → Czech: All lexical metrics including BLEU and CHRF are very biased towards ONLINE-B, with metric scores indicating that this system is better than all others by a large margin. It is ranked 7th in the human evaluation. Neural metrics and reference-free metrics are more or less correct when scoring this system. Surprisingly, ESIM is an exception to this, and also ranks ONLINE-B on top.
- Polish → English: Lexical metrics like BLEU give very low scores to ONLINE-G.
- Tamil → English: Lexical metrics consistently score ONLINE-Z above MICROSOFT_STC_INDIA, but the remaining metrics including the reference-free metrics rank them in the opposite order. The human evaluation agrees with the lexical metrics.
- Khmer → English: lexical metrics score the best system lower than the next two, whereas most neural metrics get the order of the top systems right.

5.1.5 Other Discrepancies between Metric and Human Scores

Here we briefly draw attention to particularities we spotted when manually examining the results.

- German → English: All metrics score Tohoku-AIP-NTT higher than OPPO, and UEDIN higher than PROMT_NMT.
- Russian → English: ONLINE-A, which is ranked 2nd in the human evaluation, receives low metric scores. In contrast, some metrics including BLEU and PARBLEU choose ARIEL197197, which is ranked 6th in the human evaluation, as the best system.
- Tamil → English: The highest ranked system according to human scores, GTCOM, receives lower metric scores than the next three to six systems. Metrics are biased towards ONLINE-A and against ONLINE-Z.
- Chinese → English: HUOSHAN_TRANSLATE is a clear winner according to human evaluation, but BLEU ranks it lower than the next 3 systems. The different between human scores for the next 8 systems is not statistically significant where metric ordering of the systems differently than human scores and these discrepancies aren't penalised harshly by Pearson correlation.

- English → Chinese: HUOSHAN_TRANSLATE is a clear winner according to human evaluation, but BLEU ranks it lower than the next 3 systems. The different between human scores for the next 8 systems is not statistically significant where metric ordering of the systems differently than human scores and these discrepancies aren't penalised harshly by Pearson correlation. While many metrics including BLEU have high correlations, others make major errors scoring the NIUTRANS. OPENKIWI-BERT assigns really low scores to

Overall, we note that these metric-human discrepancies often feature online systems which are probably more diverse that the MT system submissions to the WMT shared tasks.

5.1.6 Pearson vs. Kendall Tau

Overall, we found that Pearson correlation doesn't always give us the complete picture. In particular, outliers have a large influence on the correlation and can mask the presence of discrepancies between metric and human scores with the rest of the systems. But making a decision on which systems to discard is not easy.

In this paper, we also explore Kendall's Tau as an alternative to Pearson correlation. Tables 16 and 17 in the Appendix show Kendall Tau correlation of metrics over all MT systems (not including human translations).

Kendall's Tau is less sensitive to outliers, and directly measures whether metrics agree with humans when comparing pairs of systems. However, Kendall's Tau doesn't consider the differences in scores, and two metrics whose errors differ in magnitude can have the same Kendall's Tau correlation (Figure 2).

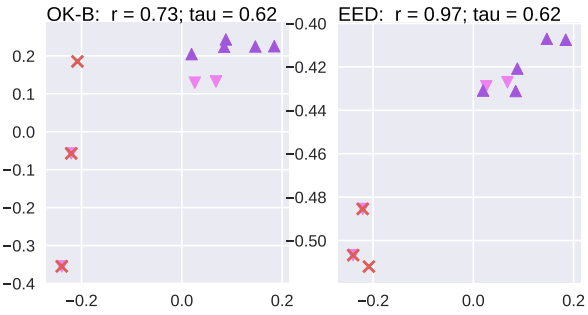


Figure 2: Scatterplots of human scores against two metrics that have the same Kendall Tau correlation with human scores, though OPENKIWI-BERT has bigger errors.

5.2 Segment and Document-Level Results

On the more fine-grained evaluation scales, PRISM and the trained neural metrics (the COMET and BLEURT family of metrics) have a better agreement with human judgements than lexical metrics

The correlations of the to-English language pairs are consistently much lower, on average, compared to that of the out-of-English language pairs. The difference could be due to the differing set of annotators: the to-English human evaluation was crowdsourced and therefore is likely to be noisier.

Finally, we find that correlations drop markedly for most language pairs if we consider only the segment/document pairs that do not contain outlier systems. We suspect that as the quality of outlier system translations is typically low, and most of the generated better-worse pairs that contain outliers can be easy for metrics. Removing these pairs would make the task a lot harder. It is also very likely that the remaining pairs of translations are noisier, which decreases metric agreement with these pairwise judgements.

The document-level correlations are typically higher than segment-level correlations. This could be due to reduced noise in human scores when averaging the scores of multiple segments. Computing metric scores over documents that contain multiple segments also helps reduce metric noise.

5.3 Reference-Based Metrics vs. Reference-Free Metrics

We have four submissions of metrics that directly compare MT outputs with the source segment: COMET-QE, OPENKIWI-BERT, OPENKIWI-XLMR, and YISI-2. Other members of the COMET family of metrics use information from both the source and reference. The remaining metrics compute scores by comparing the MT output with the reference.

While the task of comparing segments in different languages is harder than comparing segments in the same language, reference-free metrics have one advantage: they are not encumbered by reference-bias. COMET-QE is the only metric that correctly gives a high score to the human translation in German → English , and one of the few metrics that does so for English → Chinese.

This year, the reference-free metrics are highly competitive with reference-based metrics for all language-pairs. For English → Tamil, COMET-

QE which has a near perfect correlation of 0.97 even after discarding outliers. In contrast, many reference-based metrics including BLEU and chrF give really high scores to ONLINE-B, which results in low correlations.

6 Use Automatic Metrics to Detect Incorrect Human Preference

It has been argued that non-expert translators lack knowledge of translation and so might not notice subtle differences that make one translation better than another. Castilho et al. (2017) compared the evaluation of MT output of professional translators against crowd workers. Results showed that for all language pairs, the crowd workers tend to be more accepting of the MT output by giving higher fluency and adequacy scores. Toral et al. (2018) showed that the ratings acquired by professional translators show a wider gap between human and machine translations compared to judgments by non-experts. They recommend using professional linguists for MT evaluation going forward. Läubli et al. (2020) show that non-experts assess parity between human and machine translation where professional translators do not, indicating that the former neglect more subtle differences between different translation outputs. Given the previous work and the fact that the WMT human evaluation has been conducted with a mix of researchers and crowd workers, we rerun human evaluation for a subset of the submissions with professional linguists. In particular, we want to investigate if we can use the quality scores obtained by the automatic metrics to detect incorrect human ratings. We filtered out all pairs of systems where the human evaluation results disagree with all automatic metrics. Taking the metric scores as a signal, we rerun human evaluation for a subset of submissions for 2 language pairs: German→English and English→German. We hired 10 professional linguists, who rerun the source-based direct assessment human evaluation with the same document-based template that has been used for the original WMT ratings.

6.1 German→English

For German→English, we found that all automatic metrics disagree with the human evaluation results for OPPO and TOHOKU. OPPO yields a higher human rating, while all automatic metrics gave TOHOKU a higher score. To investigate which of the results to trust, we rerun the source-based direct assessment for these 2 systems with professional linguists. The results in Table 13 show that professional linguists in fact prefer the output of TOHOKU as predicted by all automatic metrics.

Evaluation	OPPO	TOHOKU
avg metric (HUMAN-A ref)	8.85	**8.95**
avg metric (Human-B ref)	10.15	**10.26**
WMT	**84.6**	81.5
z-score	**0.220**	0.179
prof. linguist	81.0	**81.7**
z-score	-0.005	**0.010**

Table 13: WMT 2020 German→English comparing the reference-based ratings acquired with crowd workers/researcher (WMT) against source-based ratings acquired with professional linguists.

6.2 English→German

For English→German, we rerun human evaluation for the top 2 ranked MT systems (based on human evaluation): OPPO, TOHOKU and the human translation HUMAN-A. The quality of human translations is usually underestimated by automatic metrics when computed with standard references. This is also visible in this year's evaluation campaign where the average metric scores of all submission for the human translation HUMAN-A is much lower when compared to the top MT submissions. To overcome this problem, Freitag et al. (2020) introduced paraphrased references that also value the translation quality of human translations and alternative (less simple/monotonic) MT output. As we can see in Table 14, the average metric scores of all submissions when computed with the paraphrased references HUMAN-P yield a much higher score for the human translation HUMAN-A when compared to all MT outputs.

The official WMT human evaluation ranked the human translation third, right behind the two MT outputs from OPPO and TOHOKU. Interestingly, based on the z-scores, WMT predicts OPPO to be of higher quality than TOHOKU which is in disagreement with most of the metric scores when calculated against both types of reference translations. Overall, the automatic metrics come to a very different ranking than the human evaluation for the top performing submissions.

Evaluation	OPPO	TOHOKU	HUMAN-A
avg metric (Human-B ref)	10.05	**10.09**	9.14
avg metric (Human-P ref)	11.93	12.07	**15.74**
WMT	87.39	**88.62**	85.10
z-score	**0.495**	0.468	0.379
prof. linguist	73.66	74.70	**84.09**
z-score	-0.051	-0.037	**0.088**

Table 14: WMT 2020 English→German comparing the source-based ratings acquired with crowd workers/researcher (WMT) against source-based ratings acquired with professional linguists.

We rerun the human evaluation with the same template, but with professional linguists. Interestingly, the human translation has been ranked first by a large margin. Furthermore, the MT output of TOHOKU has been rated as higher quality when compared to the MT output from OPPO. The results of the human evaluation with professional linguists yield a perfect correlation to the metric scores calculated with the paraphrased reference. This indicates not only the advantages of paraphrased references when scoring human translations, but also that automatic metrics can be used to identify incorrect human ratings.

7 Conclusion

This paper summarizes the results of WMT20 shared task in machine translation evaluation, the Metrics Shared Task. Participating metrics were evaluated in terms of their correlation with human judgement at the level of the whole test set (system-level evaluation), as well as at a more fine-grained level (document-level evaluation and sentences or paragraphs for segment-level evaluation). We reported scores for standard metrics requiring the reference as well metrics that compare MT output directly with the source text. For system-level, best metrics reach over 0.95 Pearson correlation or better across several language pairs. In many cases, this correlation drops considerably when the correlation is recomputed after discarding outlier systems.

Computing Pearson correlation without outliers can change the rankings of metrics, and selecting these outlier systems is not an exact science. We report results both with all systems and after discarding outliers as together, and also include Kendall Tau correlation, and hope that together, they give a more complete picture than just reporting only one of these numbers. In the end, we believe that it is impossible to adequately describe data with summary numbers, and that it's best to visualise data to understand patterns.

The results confirm the trends from previous years, namely metrics based on word or sentence-level embeddings, achieve the highest performance (Ma et al., 2018, 2019).

For some language pairs, we had two references available. On these test sets, we found that computing scores with two references rarely helped metrics achieve a higher correlation than using either reference individually. This contradicts earlier research that shows that multiple references improve correlation (Bojar et al., 2013), but is in line with more recent papers that show additional independent references might not be helpful (Freitag et al., 2020). We believe that the utility of additional independent references is dependent on the MT systems evaluated, that perhaps they are not as helpful when scoring high quality MT systems as with low/mid quality MT.

In addition to scoring MT systems, this year, we also requested scores for human reference translations. This highlighted the difference between lexical and embedding-based metrics, as lexical metrics consistently gave low scores to human translations. However, when using the English-German paraphrased references, all metrics scored the other human references above all MT systems, highlighting the advantages of using paraphrased references when scoring human translations.

In addition to human references, there are some MT systems where metrics (either the majority of metrics, or only the lexical metrics) make major errors. It remains an open question as to what it is about these systems that metrics struggle with scoring them correctly.

Compared to last year, the performance of the reference-free metrics has improved, and the correlations this year are competitive with the reference-based metrics, and in many cases, outperform BLEU. In particular, COMET-QE is good at recognising the high quality of human translations where BLEU falls short.

In terms of segment-level Kendall's τ results, the standard metrics correlations was very low for the to-English language pairs, particularly after discarding translations by outlier systems. The corre-

lations of the out-of-English language pair are more in line with recent years, reaching a maximum of above 0.6.

It has been shown that context is really important when humans are rating MT outputs (Toral et al., 2018), and the WMT human evaluation is moving towards evaluating segments with the document context (Barrault et al., 2019). This creates a mismatch with automatic metrics, all of which, this year, score each segment independently. This year, we introduce document-level evaluation of metrics. When computing document-level scores, some metrics from the COMET family include document context when computing segment scores within the document. All other metrics included in this year's evaluation either use the average of the segment scores or compute the document score based on statistics computed independently for each segment. In the future, we hope to see more metrics that consider broader context when evaluating translations at all three levels.

For this year, we are unable to draw any meaningful conclusions from the document-level evaluation task, as it is hard to tease apart the influence of noise in the ground truth, inadequate segment-level translations and inadequate translation in context of the document.

We believe that the noise in the DARR judgements is a big factor in the low correlations in the to-English language pairs. We need further research into understanding the factors that contribute to the Kendall Tau scores and how much we can trust these results.

There are shortcomings in the methods used to evaluate metrics at the system-, document-, and segment-level, and we believe that improving methods for evaluating and analysing automatic metrics is a rich area for future research.

Finally, we assume that any discrepancies between metrics and WMT manual evaluation is a metric error, and we acknowledge that this might not be true in all cases. There is always scope for improvement in human evaluation methodology, and the best practice recommendations for human evaluation are always evolving.

We rerun human evaluation by using the same template as the WMT evaluation, but switching the rater pool from non-experts to professional linguists for a subset of translations where all metrics disagree with the WMT human evaluation. This experiment revealed a new use case of automatic metrics and demonstrated that automatic metrics can be used to identify bad ratings in human evaluations. The new obtained ratings were in line with the scores suggested by the automatic metrics and also confirmed the higher translation quality of human translations when compared to MT output.

In this paper, we looked at how outliers influence metric evaluation, and we wonder how the presence of these systems influence DA annotations. In a perfect world, annotators score each translation on its own merits without being influenced by previous instances. In this world, given the presence of much worse translations, do annotators assign high scores to the remaining translations that look relatively better? Does an MT system receive an unfair advantage if it is consistently scored alongside a low-scoring outlier? And does standardising the scores of individual annotators exacerbate this issue? These and other research questions remain open this year, keeping the WMT tasks increasingly interesting as MT systems are getting closer to human performance.

Acknowledgments

Results in this shared task would not be possible without tight collaboration with organizers of the WMT News Translation Task.

We are grateful to Google for sponsoring the evaluation of selected MT systems by professional linguists. We thank all participants of the task, particularly Weiyu Wang for pointing out some inconsistencies with the original release of inputs, Jackie Lo, Ricardo Rei, and Craig Stewart for some helpful feedback and Thibault Sellam for also submitting the scores of finetuned BERT on our request.

Nitika Mathur is supported by the Australian Research Council. Ondřej Bojar would like to acknowledge the support from the Czech Science Foundation (grant n. 19-26934X, NEUREM3).

References

Manish Shrivastava Ananya Mukherjee, Hema Ala and Dipti Misra Sharma. 2020. Mee: An automatic metric for evaluation using embeddings for machine translation. (in press).

Loïc Barrault, Ondřej Bojar, Marta R Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, et al. 2019. Findings of the 2019 Conference on Machine Translation (WMT19). In

Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1), pages 1–61.

Loïc Barrault, Magdalena Biesialska, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Matthias Huck, Eric Joanis, Tom Kocmi, Philipp Koehn, Chi-kiu Lo, Nikola Ljubešić, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Santanu Pal, Matt Post, and Marcos Zampieri. 2020. Findings of the 2020 conference on machine translation (wmt20). In *Proceedings of the Fifth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Rachel Bawden, Biao Zhang, Andre Tättar, and Matt Post. 2020. ParBLEU: Augmenting metrics with automatic paraphrases for the WMT'20 metrics shared task. In *Proceedings of the Fifth Conference on Machine Translation (Volume 2: Shared Task Papers)*, Online. Association for Computational Linguistics.

Ondřej Bojar, Matouš Macháček, Aleš Tamchyna, and Daniel Zeman. 2013. Scratching the surface of possible translations. In *Text, Speech, and Dialogue*, pages 465–474, Berlin, Heidelberg. Springer Berlin Heidelberg.

Ondřej Bojar, Yvette Graham, and Amir Kamran. 2017. Results of the wmt17 metrics shared task. In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 489–513, Copenhagen, Denmark.

Ondřej Bojar, Yvette Graham, Amir Kamran, and Miloš Stanojević. 2016. Results of the WMT16 Metrics Shared Task. In *Proceedings of the First Conference on Machine Translation*, pages 199–231, Berlin, Germany.

Sheila Castilho, Joss Moorkens, Federico Gaspari, Andy Way, Panayota Georgakopoulou, Maria Gialama, Vilelmini Sosoni, and Rico Sennrich. 2017. Crowdsourcing for nmt evaluation: Professional translators versus the crowd. *Translating and the Computer*, 39.

Qian Chen, Xiaodan Zhu, Zhenhua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017. Enhanced LSTM for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, Vancouver, Canada.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Markus Dreyer and Daniel Marcu. 2012. HyTER: Meaning-equivalent semantics for translation evaluation. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 162–171, Montréal, Canada. Association for Computational Linguistics.

Markus Freitag, David Grangier, and Isaac Caswell. 2020. BLEU might be guilty but references are not innocent. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 61–71, Online. Association for Computational Linguistics.

Yvette Graham and Timothy Baldwin. 2014. Testing for significance of increased correlation with human judgment. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 172–176, Doha, Qatar. Association for Computational Linguistics.

Yvette Graham, Timothy Baldwin, and Nitika Mathur. 2015. Accurate evaluation of segment-level machine translation metrics. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1183–1191, Denver, Colorado. Association for Computational Linguistics.

Yvette Graham, Timothy Baldwin, Alistair Moffat, and Justin Zobel. 2013. Continuous measurement scales in human evaluation of machine translation. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 33–41, Sofia, Bulgaria. Association for Computational Linguistics.

Yvette Graham, Qingsong Ma, Timothy Baldwin, Qun Liu, Carla Parra, and Carolina Scarton. 2017. Improving evaluation of document-level machine translation quality estimation. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 356–361, Valencia, Spain. Association for Computational Linguistics.

Yvette Graham, Nitika Mathur, and Timothy Baldwin. 2014. Randomized significance tests in machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 266–274, Baltimore, Maryland, USA. Association for Computational Linguistics.

Boris Iglewicz and David Caster Hoaglin. 1993. *How to detect and handle outliers*, volume 16. Asq Press.

Fabio Kepler, Jonay Trénous, Marcos Treviso, Miguel Vera, and André F. T. Martins. 2019. OpenKiwi: An open source framework for quality estimation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 117–122, Florence, Italy. Association for Computational Linguistics.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395, Barcelona, Spain. Association for Computational Linguistics.

Philipp Koehn, Vishrav Chaudhary, Ahmed El-Kishky, Naman Goyal, Peng-Jen Chen, and Francisco Guzmán. 2020. Findings of the wmt 2020 shared task on parallel corpus filtering and alignment. In *Proceedings of the Fifth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Samuel Läubli, Sheila Castilho, Graham Neubig, Rico Sennrich, Qinlan Shen, and Antonio Toral. 2020. A set of recommendations for assessing human–machine parity in language translation. *Journal of Artificial Intelligence Research*, 67:653–672.

Christophe Leys, Christophe Ley, Olivier Klein, Philippe Bernard, and Laurent Licata. 2013. Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median. *Journal of Experimental Social Psychology*, 49(4):764–766.

Chi-kiu Lo. 2019. YiSi - a unified semantic MT quality evaluation and estimation metric for languages with different levels of available resources. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 507–513, Florence, Italy. Association for Computational Linguistics.

Chi-kiu Lo. 2020. Extended study on using pretrained language models and YiSi-1 for machine translation evaluation. In *Proceedings of the Fifth Conference on Machine Translation: Shared Task Papers*.

Chi-kiu Lo and Samuel Larkin. 2020. Machine translation reference-less evaluation using YiSi-2 with bilingual mappings of massive multilingual language model. In *Proceedings of the Fifth Conference on Machine Translation: Shared Task Papers*.

Qingsong Ma, Ondřej Bojar, and Yvette Graham. 2018. Results of the WMT18 metrics shared task: Both characters and embeddings achieve good performance. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 671–688, Belgium, Brussels. Association for Computational Linguistics.

Qingsong Ma, Johnny Wei, Ondřej Bojar, and Yvette Graham. 2019. Results of the WMT19 metrics shared task: Segment-level and strong MT systems pose big challenges. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 62–90, Florence, Italy. Association for Computational Linguistics.

Matouš Macháček and Ondřej Bojar. 2014. Results of the WMT14 metrics shared task. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 293–301, Baltimore, Maryland, USA. Association for Computational Linguistics.

Nitika Mathur, Timothy Baldwin, and Trevor Cohn. 2019. Putting evaluation in context: Contextual embeddings improve machine translation evaluation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2799–2808, Florence, Italy. Association for Computational Linguistics.

Nitika Mathur, Timothy Baldwin, and Trevor Cohn. 2020. Tangled up in BLEU: Reevaluating the evaluation of automatic machine translation evaluation metrics. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4984–4997, Online. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002a. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002b. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL 2002)*, pages 311–318, Philadelphia, USA.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal.

Maja Popović. 2017. chrF++: words helping character n-grams. In *Proceedings of the Second Conference on Machine Translation*, pages 612–618, Copenhagen, Denmark. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020a. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020b. Unbabel's participation in the wmt20 metrics shared task. In *Proceedings of the Fifth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Peter J Rousseeuw and Mia Hubert. 2011. Robust statistics for outlier detection. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(1):73–79.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020a. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892. Association for Computational Linguistics.

Thibault Sellam, Amy Pu, Hyung Won Chung, Sebastian Gehrmann, Qijun Tan, Markus Freitag, Dipanjan Das, and Ankur Parikh. 2020b. Learning to evaluate translation beyond english: Bleurt submissions to the wmt metrics 2020 shared task. In *Proceedings of the Fifth Conference on Machine Translation*, Online. Association for Computational Linguistics.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of the Association for Machine Transaltion in the Americas*, pages 223–231.

Peter Stanchev, Weiyue Wang, and Hermann Ney. 2019. Eed: Extended edit distance measure for machine translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 514–520, Florence, Italy. Association for Computational Linguistics.

Brian Thompson and Matt Post. 2020. Automatic machine translation evaluation in many languages via zero-shot paraphrasing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Online. Association for Computational Linguistics.

Antonio Toral, Sheila Castilho, Ke Hu, and Andy Way. 2018. Attaining the Unattainable? Reassessing Claims of Human Parity in Neural Machine Translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 113–123, Belgium, Brussels. Association for Computational Linguistics.

Tereza Vojtěchová, Michal Novák, Miloš Klouček, and Ondřej Bojar. 2019. SAO WMT19 Test Suite: Machine Translation of Audit Reports. In *Proceedings of the Fourth Conference on Machine Translation*, Florence, Italy. Association for Computational Linguistics.

Weiyue Wang, Jan-Thorsten Peter, Hendrik Rosendahl, and Hermann Ney. 2016. CharacTer: Translation edit rate on character level. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 505–510, Berlin, Germany. Association for Computational Linguistics.

Evan James Williams. 1959. *Regression analysis*. wiley.

Jin Xu, Yinuo Guo, and Junfeng Hu. 2020. Incorporate semantic structures into machine translation evaluation via ucca. In *Proceedings of the Fifth Conference on Machine Translation*, Online. Association for Computational Linguistics.

A List of Outliers

lp	Outliers
cs-en	ZLABS-NLP.1149, CUNI-DOCTRANSFORMER.1457
de-en	YOLO.1052, ZLABS-NLP.1153, WMTBIOMEDBASELINE.387
iu-en	NIUTRANS.1206, FACEBOOK_AI.729
ja-en	ONLINE-G.1564, ZLABS-NLP.66, ONLINE-Z.1640
pl-en	ZLABS-NLP.1162
ru-en	ZLABS-NLP.1164
ta-en	ONLINE-G.1568, TALP_UPC.192
zh-en	WMTBIOMEDBASELINE.183
en-cs	ZLABS-NLP.1151, ONLINE-G.1555
en-de	ZLABS-NLP.179, WMTBIOMEDBASELINE.388, ONLINE-G.1556
en-iu_news	UEDIN.1281, OPPO.722, UQAM_TANLE.521
en-iu_full	UEDIN.1281, OPPO.722, UQAM_TANLE.521
en-iu	UEDIN.1281, OPPO.722, UQAM_TANLE.521
en-pl	ONLINE-Z.1634, ZLABS-NLP.180, ONLINE-A.1576
en-ta	TALP_UPC.1049, SJTU-NICT.386, ONLINE-G.1561

Table 15: List of all MT systems that we consider as outliers

B Scatterplots

Here we show scatterplots of human and metric scores of selected metrics.

We report the correlation of each metric with human scores on all systems as well as all systems minus the outliers. Note that we do not exclude human translations when computing these correlations.

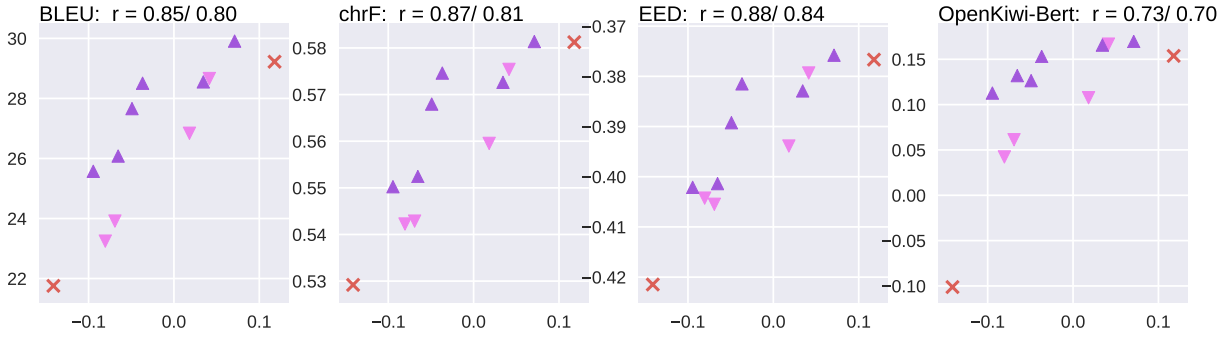
In the following scatterplots, the violet triangles indicate individual indicate MT system submissions by researchers and pink downward triangles are online systems. [8] The red crosses are outlier systems.

The black diamonds are human translations. For newstest2020 reference set, this is the HUMAN-A translation, and for newstestB2020 reference set, this is the HUMAN-B translation. The plots for English → German have two human translations included, and we annotate the label in the plot. In many cases, metric errors scoring these translations stand out.

Metric scores of MT systems with multiple references does not deviate from the scores of either reference. So we do not include the scatterplots of the other reference sets unless a human translation is included (which is interesting).

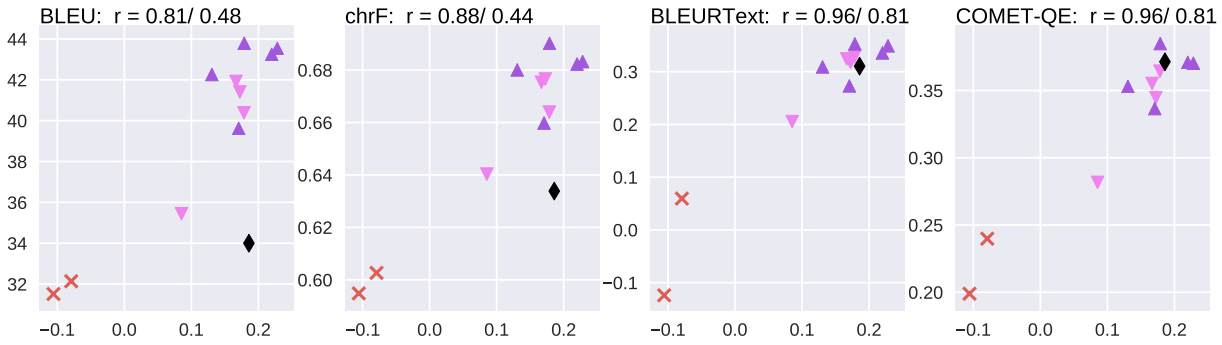
We will have scatterplots for all metrics over all reference sets in the metrics package to be made available at http://www.statmt.org/wmt20/results.html

cs-en



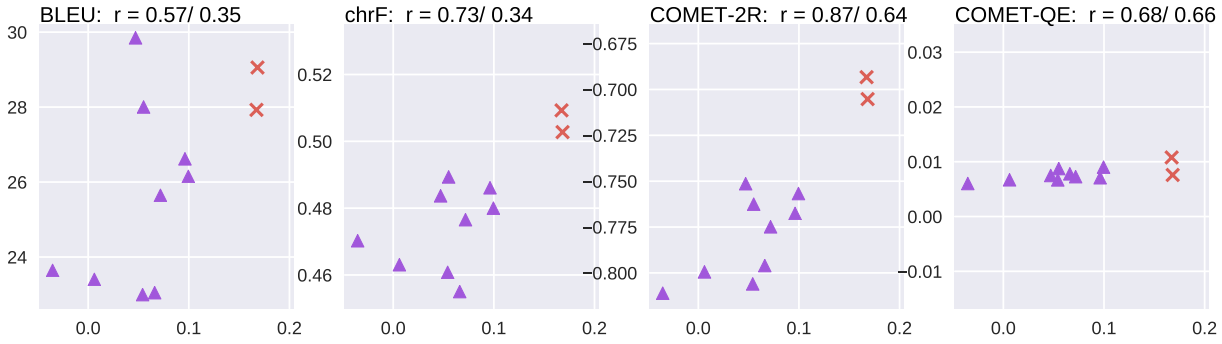
[8] We distinguish between the two in these scatterplots as we notice that metrics often make errors when scoring online systems.

de-en newstest2020[a]

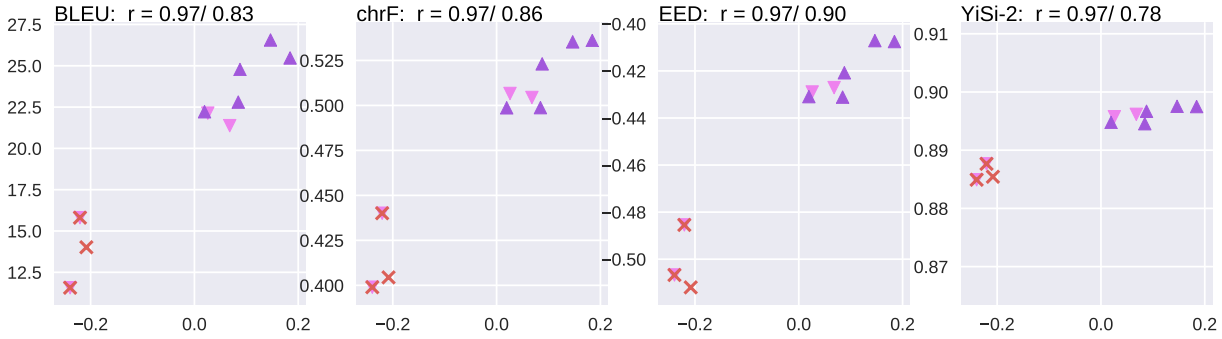


[a]Including the YOLO.1052 system, which has an extremely low quality, would make it hard to distinguish between the rest of the systems, so these plots exclude the system.

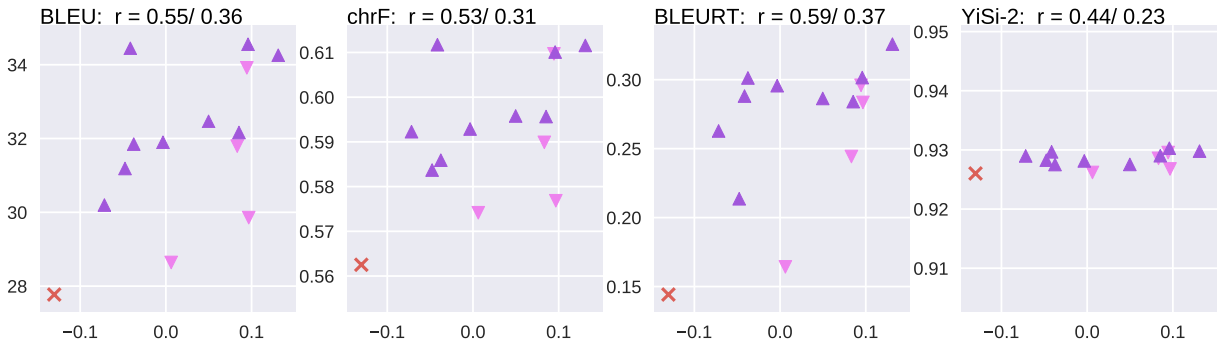
iu-en



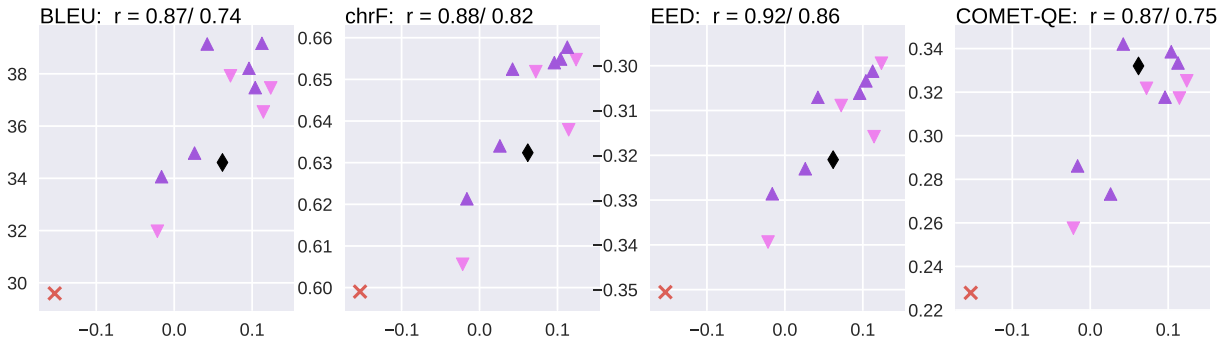
ja-en



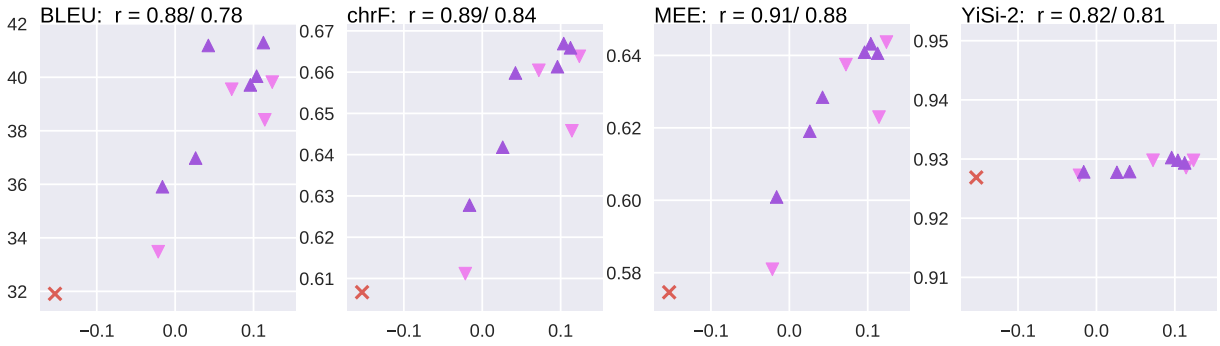
pl-en



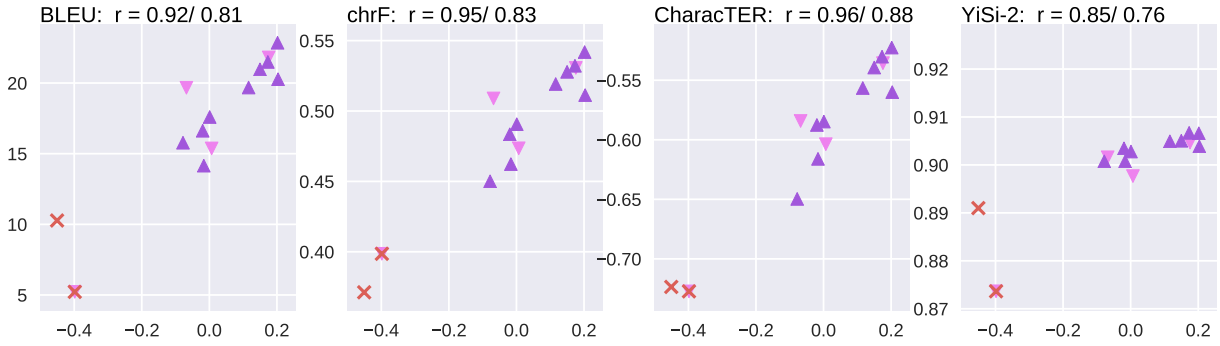
ru-en newstest2020



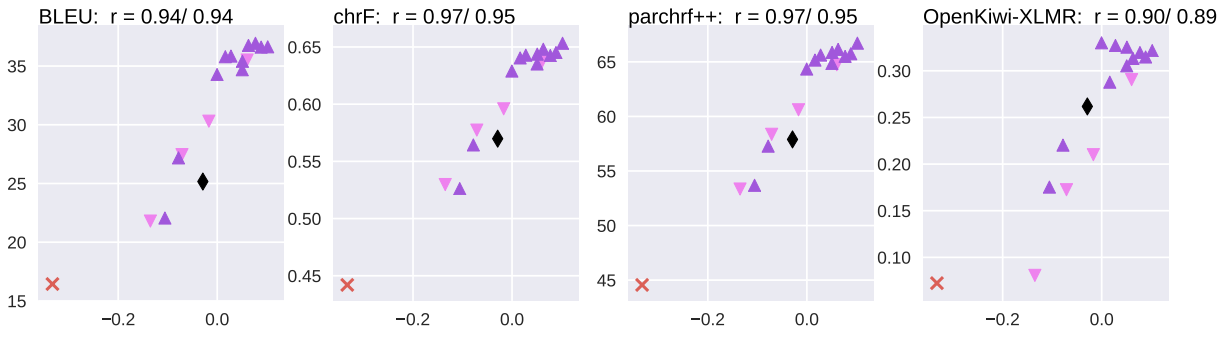
ru-en newstestB2020



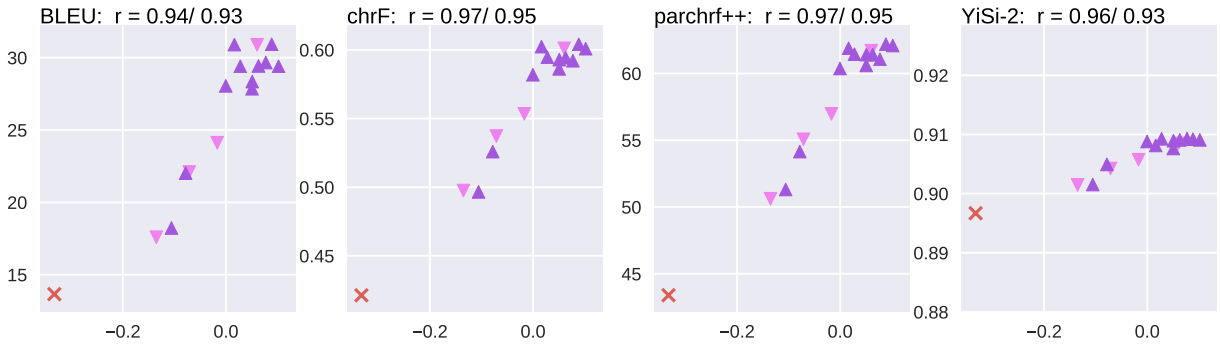
ta-en



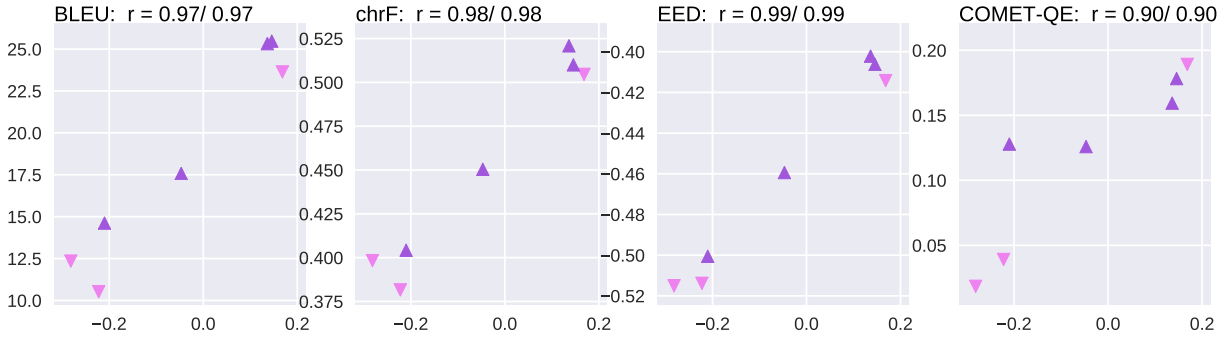
zh-en newstest2020



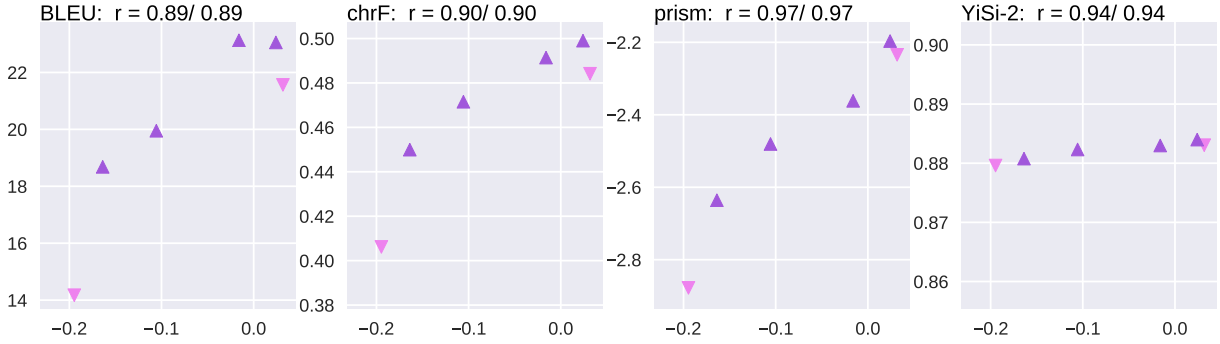
zh-en newstestB2020



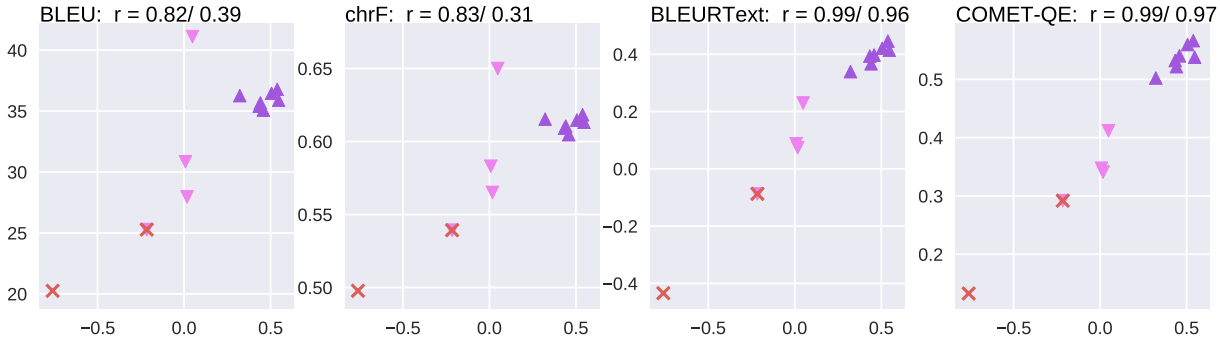
km-en



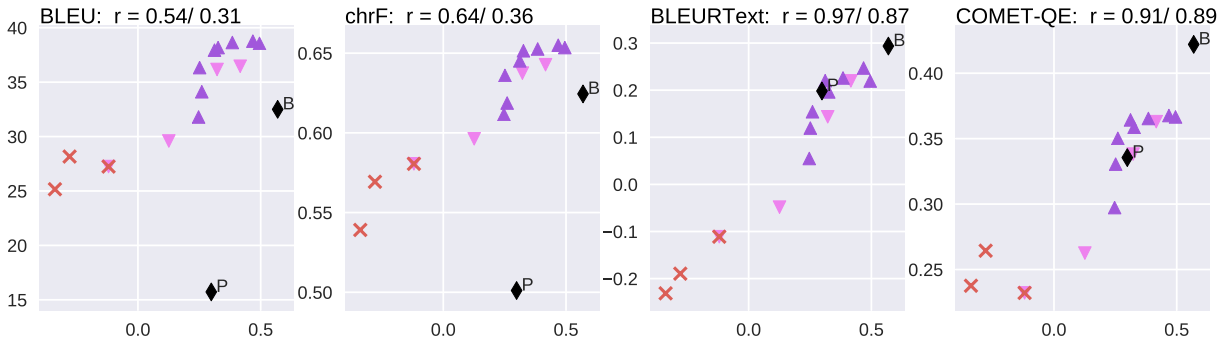
ps-en



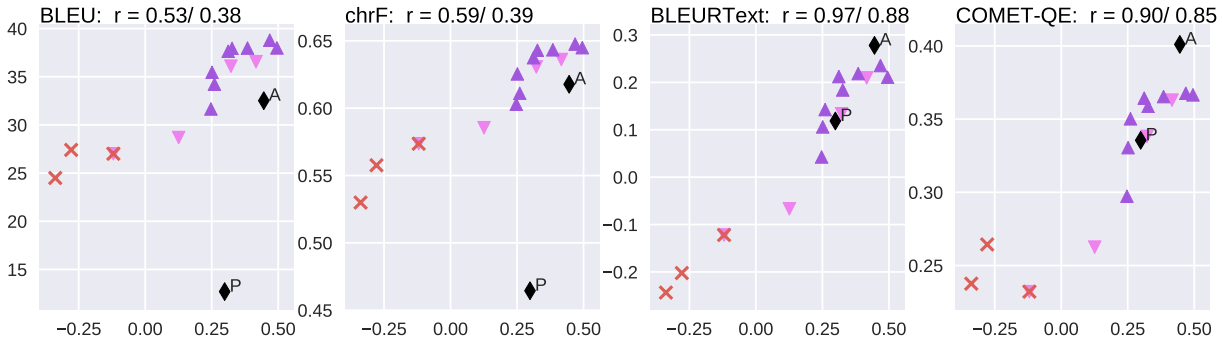
en-cs



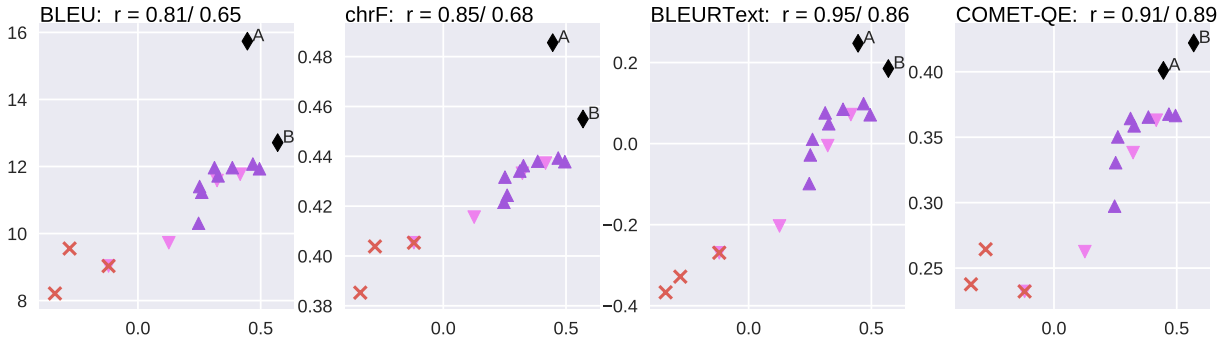
en-de newstest2020



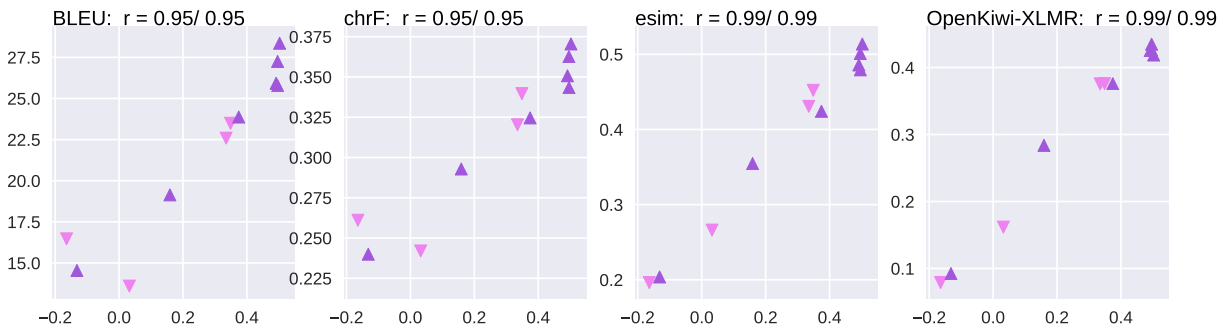
en-de newstestB2020



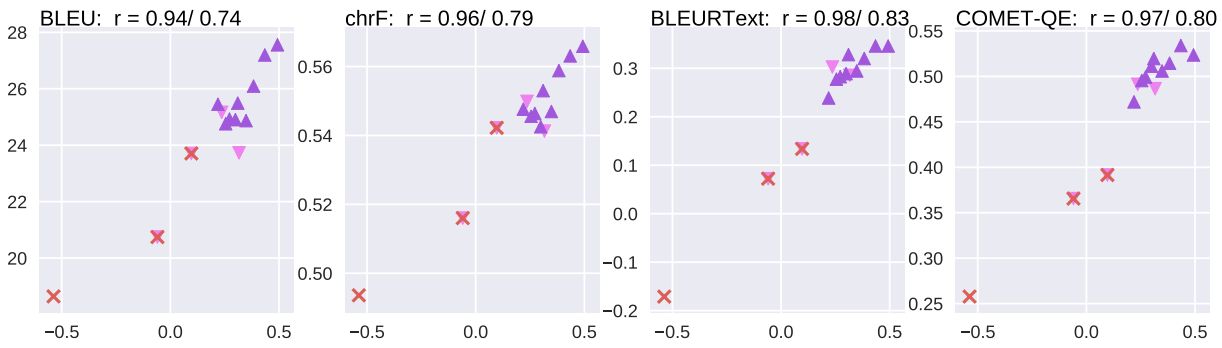
en-de newstestP2020



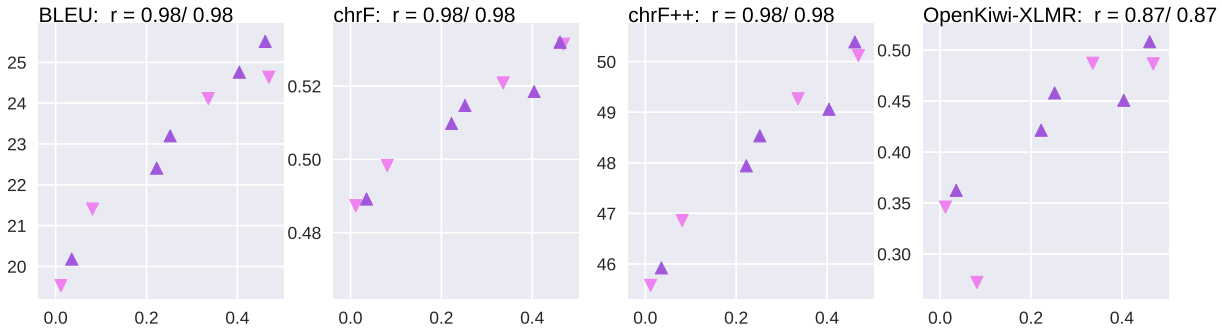
en-ja



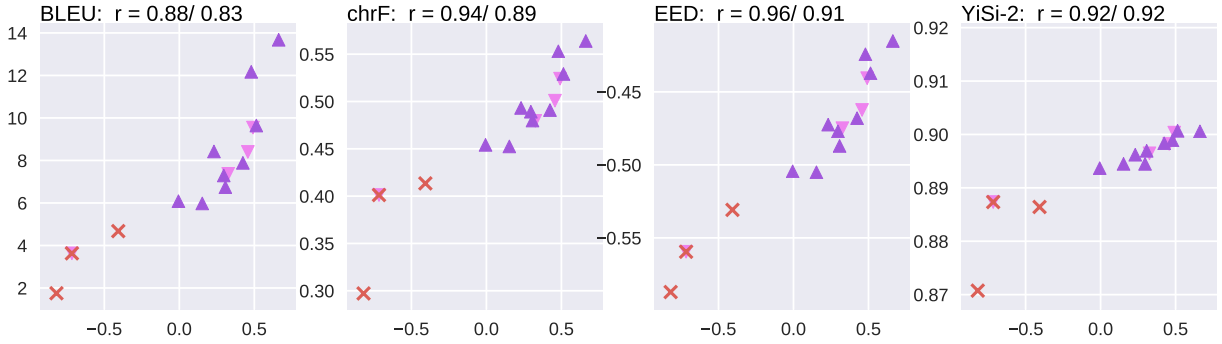
en-pl



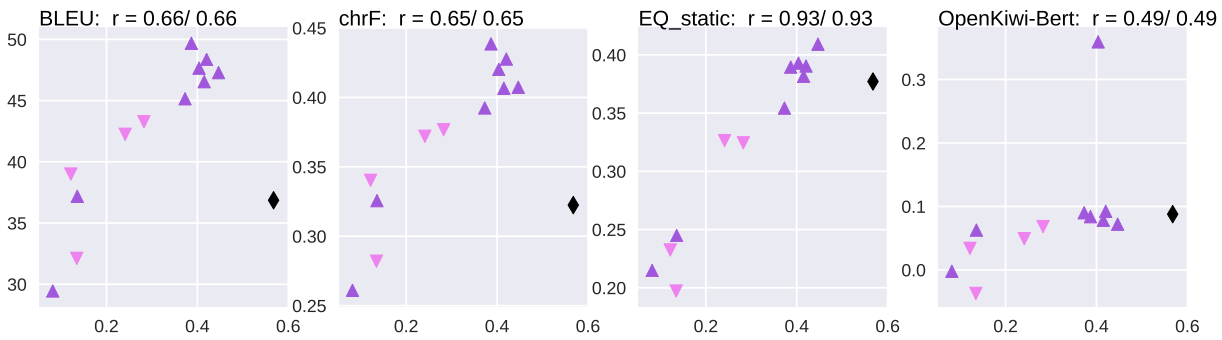
en-ru



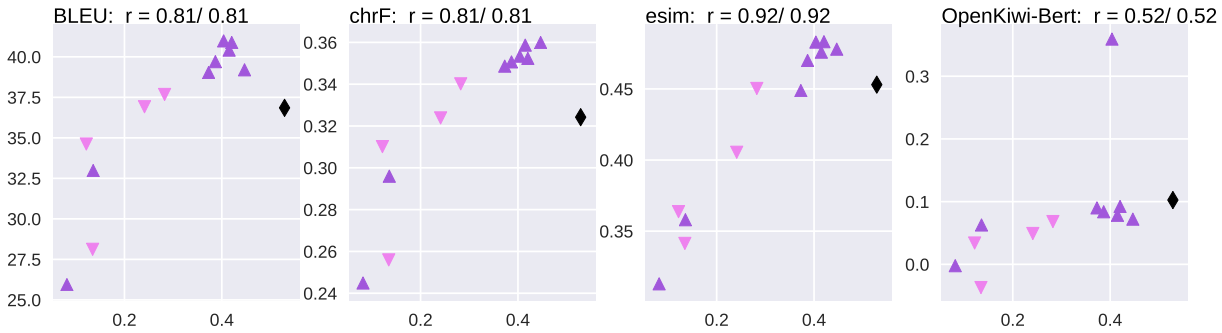
en-ta



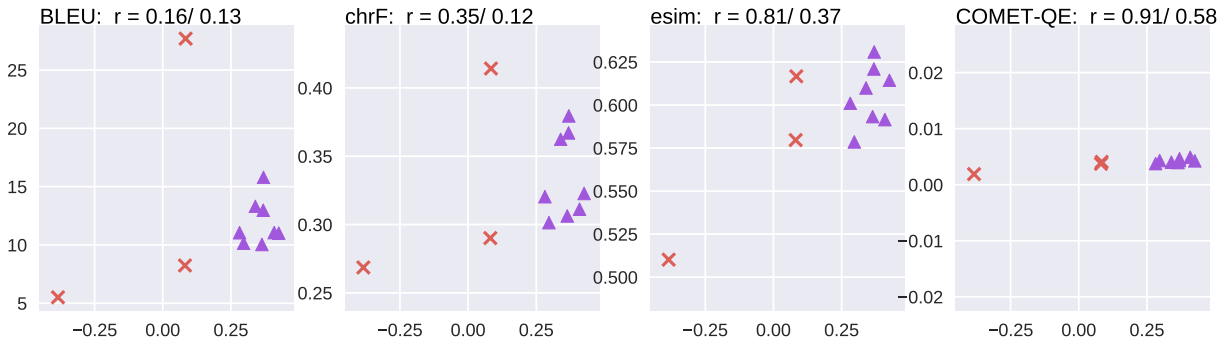
en-zh newstest2020



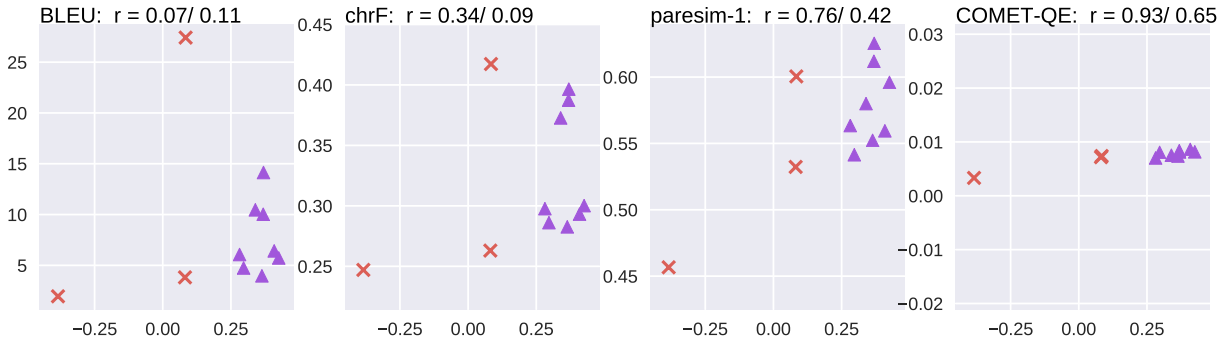
en-zh newstestB2020



en-iu Full test set



en-zh Out of domain (News) subset



C Additional System-level Results

We also report Kendall Tau correlation of metrics at the system level.

	cs-en 12	de-en 12	ja-en 10	pl-en 14	ru-en 11	ta-en 14	zh-en 16	iu-en 11	km-en 7	ps-en 6
HUMAN_RAW	0.727	0.758	0.778	0.429	0.673	0.604	0.650	0.891	0.905	1.000
SENTBLEU	0.788	0.758	0.733	0.297	0.564	0.692	0.850	0.455	0.619	0.600
BLEU	0.848	0.697	0.778	0.407	0.455	0.692	0.833	0.309	0.714	0.600
TER	0.758	0.788	0.689	0.287	0.600	0.780	0.800	0.514	0.878	0.867
CHRF++	0.818	0.697	0.778	0.407	0.673	0.714	0.850	0.418	0.619	0.733
CHRF	0.818	0.727	0.822	0.363	0.709	0.714	0.833	0.418	0.619	0.733
PARBLEU	0.809	0.779	0.778	0.420	0.491	0.685	0.807	0.404	0.714	0.867
PARCHRF++	0.818	0.727	0.822	0.407	0.709	0.714	0.817	0.491	0.619	0.733
CHARACTER	0.758	0.758	0.822	0.341	0.745	0.692	0.800	0.527	0.810	0.733
EED	0.788	0.727	0.733	0.297	0.782	0.758	0.833	0.636	0.714	0.733
YISI-0	0.758	0.758	0.689	0.231	0.782	0.802	0.833	0.600	0.714	0.733
SWSS+METEOR	–	–	0.822	0.341	0.818	0.736	0.817	0.491	0.714	0.733
MEE	0.758	0.697	0.867	0.363	0.709	0.692	0.783	0.636	0.714	0.733
PRISM	0.758	0.727	0.867	0.341	0.564	0.648	0.800	0.673	0.714	0.867
YISI-1	0.758	0.758	0.778	0.451	0.564	0.692	0.817	0.673	1.000	0.867
BERT-BASE-L2	0.758	0.848	0.822	0.407	0.491	0.604	0.633	0.564	1.000	0.867
BERT-LARGE-L2	0.758	0.848	0.867	0.341	0.564	0.626	0.700	0.527	1.000	0.867
MBERT-L2	0.758	0.818	0.822	0.429	0.564	0.604	0.750	0.673	1.000	0.867
BLEURT	0.758	0.788	0.822	0.407	0.600	0.604	0.650	0.527	1.000	0.867
BLEURT-EXTENDED	0.727	0.848	0.778	0.341	0.455	0.582	0.617	0.527	0.905	0.867
ESIM	0.727	0.848	0.822	0.451	0.491	0.670	0.717	0.636	1.000	0.867
PARESIM-1	0.727	0.879	0.822	0.451	0.491	0.670	0.700	0.636	1.000	0.867
COMET	0.727	0.758	0.778	0.407	0.564	0.626	0.733	0.636	1.000	0.867
COMET-2R	0.727	0.788	0.778	0.451	0.527	0.582	0.717	0.600	1.000	0.867
COMET-HTER	0.667	0.788	0.822	0.275	0.491	0.604	0.533	0.564	1.000	0.867
COMET-MQM	0.667	0.727	0.822	0.275	0.455	0.582	0.517	0.636	1.000	1.000
COMET-RANK	0.576	0.727	0.822	0.341	0.455	0.626	0.650	0.309	0.810	1.000
BAQ_DYN	–	–	–	–	–	–	0.817	–	–	–
BAQ_STATIC	–	–	–	–	–	–	0.867	–	–	–
COMET-QE	0.697	0.788	0.778	0.297	0.455	0.516	0.550	0.491	0.905	0.733
OPENKIWI-BERT	0.697	0.667	0.733	0.187	0.455	0.429	0.450	-0.055	0.714	0.467
OPENKIWI-XLMR	0.727	0.636	0.822	0.275	0.418	0.560	0.567	0.018	1.000	0.867
YISI-2	0.576	0.515	0.778	0.319	0.527	0.582	0.750	0.491	0.810	0.867

Table 16: Kendall Tau correlation of system-level metrics with DA human assessment for all MT systems not including Human translations. In addition to the metrics, we also include raw human scores where annotator scores were not standardised.

	en-cs 12	en-de 14	en-ja 11	en-pl 14	en-ru 9	en-ta 15	en-zh 12	en-iu_full 11	en-iu_news 11
HUMAN_RAW	1.000	0.868	0.964	0.846	0.778	0.810	0.818	0.600	0.600
SENTBLEU	0.515	0.802	0.855	0.604	0.944	0.867	0.727	0.236	0.273
BLEU	0.515	0.802	0.818	0.582	0.889	0.829	0.727	0.236	0.236
TER	0.515	0.824	0.018	0.641	0.556	0.752	0.242	0.309	0.309
CHRF++	0.485	0.868	0.782	0.604	0.889	0.829	0.727	0.309	0.309
CHRF	0.485	0.868	0.818	0.604	0.889	0.810	0.727	0.345	0.309
PARBLEU	0.504	0.736	0.611	0.633	0.761	0.842	0.718	0.404	0.345
PARCHRF++	0.515	0.846	0.818	0.670	0.889	–	0.727	–	–
CHARACTER	0.515	0.890	0.782	0.560	0.944	0.771	0.697	0.236	0.345
EED	0.545	0.868	0.782	0.604	0.833	0.867	0.727	0.273	0.273
YISI-0	0.545	0.846	0.818	0.604	0.944	0.790	0.515	0.236	0.345
MEE	0.576	0.802	–	0.582	0.667	0.829	–	0.273	0.382
PRISM	0.818	0.868	0.818	0.670	0.611	0.562	0.576	0.418	0.600
YISI-1	0.606	0.868	0.782	0.626	0.833	0.810	0.758	0.091	0.273
YISI-COMBI	–	0.824	–	–	–	–	–	–	–
BLEURT-YISI-COMBI	–	0.824	–	–	–	–	–	–	–
MBERT-L2	0.788	0.846	0.782	0.736	0.778	0.752	0.909	–	–
BLEURT-EXTENDED	0.879	0.802	0.782	0.780	0.833	0.771	0.848	0.382	0.345
ESIM	0.606	0.912	0.855	0.692	0.833	0.752	0.788	0.382	0.455
PARESIM-1	0.667	0.890	0.818	0.692	0.833	0.752	0.818	0.382	0.455
COMET	0.909	0.846	0.745	0.736	0.722	0.771	0.606	0.382	0.382
COMET-2R	0.909	0.890	0.891	0.714	0.611	0.790	0.606	0.309	0.418
COMET-HTER	0.909	0.802	0.818	0.736	0.667	0.619	0.576	0.491	0.491
COMET-MQM	0.909	0.802	0.818	0.736	0.667	0.619	0.545	0.527	0.455
COMET-RANK	0.848	0.780	0.782	0.692	0.556	0.524	0.515	0.127	0.345
BAQ_DYN	–	–	–	–	–	–	0.697	–	–
BAQ_STATIC	–	–	–	–	–	–	0.788	–	–
EQ_DYN	–	–	–	–	–	–	0.727	–	–
EQ_STATIC	–	–	–	–	–	–	0.818	–	–
COMET-QE	0.848	0.802	0.709	0.802	0.667	0.543	0.576	0.600	0.673
OPENKIWI-BERT	0.758	0.780	0.236	0.538	0.722	0.314	0.606	-0.273	0.200
OPENKIWI-XLMR	0.909	0.780	0.818	0.692	0.667	0.657	0.545	0.018	0.200
YISI-2	0.485	0.582	0.527	0.077	0.444	0.886	0.121	0.309	0.455

Table 17: Kendall Tau correlation of out-of-English system-level metrics with DA human assessment for all MT systems not including Human translations. In addition to the metrics, we also include raw human scores where annotator scores were not standardised.

Findings of the WMT 2020 Shared Task on Parallel Corpus Filtering and Alignment

Philipp Koehn[*†], **Vishrav Chaudhary**[†], **Ahmed El-Kishky**[†]
Naman Goyal[†], **Peng-Jen Chen**[†], **Francisco Guzmán**[†]
phi@jhu.edu, vishrav@fb.com, ahelk@fb.com
naman@fb.com, pipibjc@fb.com, fguzman@fb.com
[*]Johns Hopkins University, Baltimore, Maryland, United States
[†]Facebook AI, Menlo Park, California, United States

Abstract

Following two preceding WMT Shared Tasks on Parallel Corpus Filtering (Koehn et al., 2018, 2019), we posed again the challenge of assigning sentence-level quality scores for very noisy corpora of sentence pairs crawled from the web, with the goal of sub-selecting the highest-quality data to be used to train machine translation systems. This year, the task tackled the low resource condition of Pashto–English and Khmer–English and also included the challenge of sentence alignment from document pairs. 10 participants from companies, national research labs, and universities participated in this task.

1 Introduction

The field of Machine Translation has experienced significant advances in recent years thanks to improvements in neural modeling (Bahdanau et al., 2015; Gehring et al., 2016; Vaswani et al., 2017), as well as the availability of large parallel corpora for training (Tiedemann, 2012; Smith et al., 2013; Bojar et al., 2017). Unfortunately, today's neural machine translation models, perform poorly on *low-resource* language pairs, for which clean, high-quality training data is lacking (Koehn and Knowles, 2017). Improving performance on low resource language pairs has high impact considering that these languages are spoken by a large fraction of the world population. This is a particular challenge for industrial machine translation systems that need to support hundreds of languages in order to provide adequate services to their multilingual user base.

While there have been advances in using monolingual corpora (Lample et al., 2018; Liu et al., 2020) and parallel corpora in multiple language pairs (Aharoni et al., 2019; Fan et al., 2020), the best training data for machine translation are still parallel corpora in the targeted language pair and domain.

Parallel corpora are typically gathered from any available source without much guarantees about quality. This is especially the case for parallel corpora that are extracted from the web without much control over which web sites are mined. Since noisy training data has been recognized as a challenge for neural machine translation training (Khayrallah and Koehn, 2018), an essential step in using such data is filtering or discounting noisy sentence pairs.

Recently, there is increased interest in the filtering of noisy parallel corpora to improve the data that can be used to train translation systems. The Shared Task on Parallel Corpus Filtering and Alignment at the Conference for Machine Translation (WMT 2020) was organized to promote research to make learning from noisy data more viable for low-resource languages. It is similar to the previous year's task but tackles different languages (Pashto and Khmer instead of Nepali and Sinhala) and also included the challenge to extract sentence pairs from document pairs.

The shared task is organized similarly to previous years (Koehn et al., 2018, 2019). We provide about 11.6 million word noisy parallel data for Pashto-English and 58.3 million word noisy parallel data for Khmer-English. We also provide small amounts of clean parallel data of varying quality and monolingual data from Wikipedia and CommonCrawl.

Participants developed methods to assign a quality score for each sentence pair. These scores are used to filter the web crawled corpora down to

Proceedings of the 5th Conference on Machine Translation (WMT), pages 726–742

a fixed size (5 million English words), train neural machine translation systems on these subsets, and measure their quality with the BLEU score on a test set of multi-domain Wikipedia content (Guzmán et al., 2019).

This paper gives an overview of the task, presents the results for the participating systems and provides analysis on additional subset sizes, the average sentence length of sub-selected data, and overlap between the submissions.

2 Related Work

Although the idea of crawling the web indiscriminately for parallel data goes back to the 20th century (Resnik, 1999), work in the academic community on extraction of parallel corpora from the web has so far mostly focused on large stashes of multilingual content in homogeneous form, such as the Canadian Hansards, Europarl (Koehn, 2005), the United Nations (Rafalovitch and Dale, 2009; Ziemski et al., 2015), or European Patents (Täger, 2011). A nice collection of the products of these efforts is the OPUS web site[1] (Tiedemann, 2012).

2.1 Parallel Corpus Acquisition

Noisy parallel documents and parallel sentences were sourced from the CCAligned[2] dataset (El-Kishky et al., 2020a), a massive collection of cross-lingual web documents covering over 8k language pairs aligned from 68 Common Crawl snapshots. Additional parallel data was sourced from the Paracrawl project – a large-scale effort to crawl text from the web[3] (Bañón et al., 2020).

Acquiring parallel corpora from the web (El-Kishky et al., 2020b) is an active area of research that typically involves identifying web sites with parallel text, downloading the documents from the web site, aligning document pairs (Buck and Koehn, 2016; Thompson and Koehn, 2020; El-Kishky and Guzmán, 2020), and aligning sentence pairs. A final stage of the processing pipeline filters out non-parallel sentence pairs. Such noise exists either because the original web site did not have any actual parallel data (garbage in, garbage out), only partially-parallel data, or due to failures of processing steps.

[1] http://opus.nlpl.eu
[2] http://statmt.org/cc-aligned
[3] http://www.paracrawl.eu/

2.2 Sentence Alignment

Sentence alignment has been a very active field of research since the early days of statistical machine translation. An influential early method is based on sentence length, measured in words (Gale and Church, 1993). Several researchers proposed including lexical information (Chen, 1993; Moore, 2002) with the emergence of tools that use provided bilingual dictionaries (Varga et al., 2005) or acquire them during in an unsupervised fashion (Braune and Fraser, 2010). Later work introduced scoring methods that use MT to get both documents into the same language (Sennrich and Volk, 2010) or use pruned phrase tables from a statistical MT system (Gomes and Pereira Lopes, 2016). Both methods anchor high-probability 1–1 alignments in the search space and then fill in and refine alignments. More recently, Thompson and Koehn (2019) introduced the use of sentence embeddings and a coarse-to-fine search method to the task (Vecalign).

2.3 Filtering Noisy Parallel Corpora

In 2016, a shared task on sentence pair filtering[4] was organized, albeit in the context of cleaning translation memories which tend to be cleaner than the data at the end of a pipeline that starts with web crawls.

There is a robust body of work on filtering out noise in parallel data. For example: Taghipour et al. (2011) use an outlier detection algorithm to filter a parallel corpus; Xu and Koehn (2017) generate synthetic noisy data (inadequate and non-fluent translations) and use this data to train a classifier to identify good sentence pairs from a noisy corpus; and Cui et al. (2013) use a graph-based random walk algorithm and extract phrase pair scores to weight the phrase translation probabilities to bias towards more trustworthy ones.

Most of this work was done in the context of statistical machine translation, but more recent work targets neural models. Carpuat et al. (2017) focus on identifying semantic differences in translation pairs using cross-lingual textual entailment and additional length-based features, and demonstrate that removing such sentences improves neural machine translation performance.

As Rarrick et al. (2011) point out, one type of noise in parallel corpora extracted from the web

[4] NLP4TM 2016: Shared task http://rgcl.wlv.ac.uk/nlp4tm2016/shared-task/

are translations that have been created by machine translation. Venugopal et al. (2011) propose a method to watermark the output of machine translation systems to aid this distinction, with a negligible loss of quality. Antonova and Misyurev (2011) report that rule-based machine translation output can be detected due to certain word choices, and statistical machine translation output can be detected due to lack of reordering. It is notable that none of the participants in our shared task have tried to detect machine translation.

There is a rich literature on data selection which aims at sub-sampling parallel data relevant for a task-specific machine translation system (Axelrod et al., 2011). Van der Wees et al. (2017) find that the existing data selection methods developed for statistical machine translation are less effective for neural machine translation. This is different from our goals of handling noise since those methods tend to discard perfectly fine sentence pairs that are just not relevant for the targeted domain. Our task is focused on data quality that is relevant for all domains.

2.4 Impact of Noise on Neural Machine Translation

Belinkov and Bisk (2017) investigate the impact of noise on neural machine translation. They focus on creating systems that can *translate* the kinds of orthographic errors (typos, misspellings, etc.) that humans can comprehend. In contrast, Khayrallah and Koehn (2018) examine noisy *training* data and focus on types of noise occurring in web-crawled corpora. They carried out a study about how noise that occurs in crawled parallel text impacts statistical and neural machine translation.

Neural machine translation model training may combine data selection and model training, taking advantage of the increasing quality of the model to better detect noisy data or to increasingly focus on cleaner parts of the data (Wang et al., 2018; Kumar et al., 2019).

2.5 Findings of Previous Shared Tasks

We organized versions of this shared task in the previous two years. In 2018, we started with a high-resource language pair (German–English) and a very large web-crawled parallel corpus, a subset of the Paracrawl corpus consisting of 1 billion English words (Koehn et al., 2018). The best-performing submission (Junczys-Dowmunt, 2018) used neural machine translation systems in both translation directions to score sentence pairs with dual cross-entropy.

Last year, we moved the focus to low resource languages (Koehn et al., 2019) with smaller noisy parallel corpora, comprising 50-60 million words for Nepali–English and Sinhala–English. For these languages much less clean parallel data was available and hence many of the methods developed for high-resource languages are less reliable. The best-performing submission that year (Chaudhary et al., 2019) also considered dual cross-entropy but found that matching multilingual sentence embeddings (Schwenk, 2018) gave better results.

2.6 Monolingual Pre-Training

By now, neural machine translation systems are rarely trained only on the parallel corpus of the desired language pair. Common foundations are pre-trained models trained on multiple language pairs which share the source or target language (Aharoni et al., 2019; Fan et al., 2020) or monolingual pre-training methods (Liu et al., 2020). Often, the models are also improved by a second stage of training that uses back-translated synthetic parallel data that was generated from first stage model — a process that may be iterated (Hoang et al., 2018).

To reflect such a more realistic training setup, we provided pre-trained models that were trained on monolingual data using a denoising auto-encoder method called mBART (Liu et al., 2020). Here, monolingual data is converted into input and output pairs by (a) masking out words in the input, forcing the model to learn the correct word or word sequence from the context, and (b) shuffling the order of a few concatenated sentence pairs.

3 Shared Task Definition

The shared task tackled the problem of filtering parallel corpora. Given a noisy parallel corpus (crawled from the web), participants developed methods to align sentences in document pairs and to filter it to a smaller size of high quality sentence pairs.

3.1 Filtering

For the filtering-only task, we provided a very noisy 58.3 million word corpus for Khmer–English (English token count) and a 11.6 million word corpus for Pashto–English, crawled from the

web (see Section 4.3 for details). We asked participants to generate sentence-level quality scores that allow selecting subsets of sentence pairs that amount to 5 million words, counted on the English side. This amount was chosen based on preliminary experiments (we report below on additional subset sizes).

Participants in the shared task submitted a file with quality scores, one score per line, corresponding to the sentence pairs. Scores are only required to have the property that higher scores indicate better quality. The scores were uploaded to a Google Drive folder which remains publicly accessible.[5]

3.2 Alignment

We also released the document pairs from which we extracted the sentence pairs. For Khmer–English, we released 391,250 document pairs, for Pashto-English 45,312 document pairs.

Participants were encouraged to develop novel methods for sentence alignment. The resulting sentence pairs also had to be annotated with quality scores, as in the filtering-only tasks, and uploaded with quality scores to the same Google Drive folder.

3.3 Evaluation

The submissions were scored by building a neural machine translation system (Ott et al., 2019) trained on this data, and then measuring their BLEU score on the `flores` Wikipedia test sets (Guzmán et al., 2019). The neural machine translation model was either randomly initialized or initialized by monolingual pre-training (mBART).

For development purposes, we released configuration files and scripts that mirror the official testing procedure with a development test set. The development pack consists of:

- A script to subsample corpora based on quality scores.
- `fairseq` scripts to train and test a neural machine translation system.
- A pre-trained mBART model for continued training.
- The `flores-dev` set of Wikipedia translations as development set.
- The `flores-devtest` set of Wikipedia translations as development test set.

[5] https://bit.ly/2IoOXOr

Corpus	Sentence Pairs	English Words
Pashto-English GNOME	95,312	277,188
KDE4	3,377	8,881
Tatoeba	31	239
Ubuntu	9,645	26,626
Bible	13,432	298,522
TED Talks	664	11,157
Wikimedia	737	37,566

Table 1: Provided clean parallel data for Pashto–English.

The web site for the shared task[6] provided detailed instructions on how to use these tools to replicate the official testing environment.

4 Data

We provided three types of data for this shared task: (1) clean parallel and monolingual data, including related language data in Hindi, to train models that aid with the filtering task, (2) the noisy parallel data crawled from the web which participants have to score for filtering, and (3) development and test sets that are used to evaluate translation systems trained on filtered data.

4.1 Clean Parallel Data

For Pashto (see Table 1 for detailed statistics), the largest data sets are the Bible (prepared for us by Arya McCarthy and David Yarowsky), various data sets from OPUS[7] (GNOME, KDE4, and Unbuntu software localization; Tatoeba volunteer translations; and Wikimedia), and a TED Talks corpus created for this task, crawled from TED web site, and sentence-aligned with Vecalign (Thompson and Koehn, 2019).

For Khmer (see Table 2 for detailed statistics), the largest data sets are the alignment of 2 English with 4 Khmer Bibles, various data sets from OPUS (GNOME, KDE4, and Ubuntu software localization; GlobalVoices citizen journalism articles; Tatoeba volunteer translations; and Wikimedia). We also re-aligned the Jehova's Witness corpus (JW300), a collection of religious texts, with Vecalign.

[6] http://www.statmt.org/wmt20/parallel-corpus-filtering.html

[7] http://opus.nlpl.eu/

Corpus	Sentence Pairs	English Words
GNOME	56	233
GlobalVoices	793	14,294
KDE4	120,087	767,919
Tatoeba	748	3,491
Ubuntu	6,987	27,413
Bible	54,222	1,176,418
JW300	107,156	1,827,348

Table 2: Provided clean parallel data for Khmer–English.

	Wikipedia	CommonCrawl
Pashto	76,557	6,558,180
Khmer	132,666	13,832,947
English	67,796,935	1,806,450,728

Table 3: Provided clean monolingual data (number of sentences).

For both language pairs, the available clean parallel data is rather small and mostly out-of-domain. It is not sufficient to build reasonable machine translation systems. In fact, even the provided raw unfiltered noisy parallel data gives better results when used directly for training.

4.2 Clean Monolingual Data

Monolingual data is always available in much larger quantities, and we provided data from two sources: Wikipedia and CommonCrawl. Both contain language that is similar to what is expected in the noisy web data to be filtered.

We filtered the data to eliminate overlap with the development and test sets. See Table 3 for detailed statistics.

4.3 Noisy Parallel Data

Noisy parallel data sourced from CCAligned and Paracrawl follow different philosophies. While CCAligned mines bitexts from a high-precision set of aligned web-documents yielding cleaner parallel bitexts, the noisy parallel corpora from Paracrawl are the outcome of a processing pipeline aimed at high recall at the cost of precision, yielding noisy bitexts. They exhibit noise of all kinds: wrong language in source and target, sentence pairs that are not translations of each other, bad language (incoherent mix of words and non-words), incomplete or bad translations, etc.

To ensure that CCAligned yields additional noisy pairs, we don't perform any filtering after mining bitexts from the CCAligned corpus.

We used the processing pipeline of the Paracrawl project to create the data, using the clean parallel data to train underlying models such as the dictionary used by Hunalign (Varga et al., 2007) and a statistical translation model used by the document aligner. The provided parallel corpus is the raw output of the crawling pipeline, with sentence pairs de-duplicated but otherwise no further filtering performed. See Table 4 for statistics of the corpus and Tables 5 and 6 for some example sentences.

4.4 Development and Test Sets

For test and development purposes, we use the `flores` Wikipedia data sets (Guzmán et al., 2019). These sets are multi-domain, that is they were sampled from Wikipedia documents with a diverse set of topics. In Table 7 we present the statistics of these sets. The official scoring of machine translation systems generated from the sub-sampled data sources is done on the *test* set.

5 Evaluation Protocol

The testing setup mirrors the development environment that we provided to the participants.

5.1 Participants

We received submissions from 10 different organizations, and an additional baseline LASER submission that was posted on the website. See Table 8 for the complete list of participants. The participant's organizations are quite diverse, with 3 participants from the United States, 2 participants from China, and 1 participant each from Canada, Egypt, Turkey/China, Scotland, and Spain. 3 of the participants are universities, 4 are companies, 1 is a joint company/university participant, and 2 are national research organizations. There was little participant overlap between this year's shared task and last year's shared task. Only AFRL and NRC participated also last year.

Each participant submitted up to 3 different sets of scores, not all participants addressed both languages, resulting in a total of 16 different submissions for Pashto and 11 different submissions for Khmer, including a baseline submission of using

	Sentence Pairs	English Words	Document Pairs
Pashto–English	1,022,883	11,551,009	45,312
Khmer–English	4,169,574	58,347,212	391,250

Table 4: Noisy parallel data to be filtered (de-duplicated raw output). Data is made available as aligned sentence pairs (see table for number of English words) and as document pairs for which sentence alignment has to be performed.

Pashto	English
د Mikoyan-Gurevich MiG-29 (ناټو کوډ مګ‌ځنۍ‌ای) د شوروي اتحاد د هوايي ځلبي لپاره لومړی په 1970s کې جوړ او د چ لومړنی الوتنه ترسره کـړه په 6، 1977 کې چې په 1983 کې د شوروي پوځ د خدمتونو ته ننوتل او اوس هم د روسيې د هوايي ځواک او نورو له خوا نړ کـارول. 1,100 زیات نسخې تر اوسه جوړ شوې دي.	The Mikoyan-Gurevich MiG-29 (NATO code Fulcrum) is a fighter aircraft of the Soviet air supremacy developed in the early 1970s and whose first flight took place October 6, 1977. It entered service in the Soviet army in 1983 and is still used today by the Russian Air Force and many others. More than 1,100 copies have so far been built.
تشناب شاورونه جوړونکي لپاره د یونان عرضه کوونکی (1)	Bathroom faucets manufacturer supplier wholesale for Yerevan (1)
تیلیفون: 4663278-0543	Telefon: 0543-4663278
بې‌اکـتنه شوې: تیموتی شیم	Reviewed by: Timothy Shim
ستاسو IP پته 54.227.76.35 ده. My-ip-is.com د لیدلو لپاره کار کیدی شي IP پتهد موندلو لپاره جیو ټایمز د آی IP پته، د پراکسي کشف، د بریښنالیک بریښنالیک او د تور لیست چکونه. نوی: زموږ سره د انټرنیټ چټک وګورئ چټک ټسټ. غواړئ خپل پی پی رومن شمیري په اړه پوه شئ؟ خپل چک وګورئ د رومن شمیره IP.	Your IP address is 54.227.76.35. My-ip-is.com can be handy for looking up IP addresses, to find out the GeoLocation of a IP address, proxy detection, email tracing and blacklist checks. New: Check your Internet Speed with our Speed Test. Want to know your IP in Roman Numerals? Check Your Roman Numerals IP.

Table 5: Examples of relatively good sentence pairs from the noisy corpus for Pashto–English. Note that unreliable sentence splitting for Pashto led to merging of sentence pairs.

Khmer	English
21:13 និងព្រះយេស៊ូវចូលមកជិត, ហើយគាត់បានយកនំប៉័ង, ហើយគាត់បានផ្តល់ឱ្យវាដើម្បីឱ្យពួកគេ, ដូចគ្នានេះដែរជាមួយនឹងត្រី.	21:13 And Jesus approached, and he took bread, and he gave it to them, and similarly with the fish.
ឌាងយ៉ូអាន់ • ខែមករា 8, 2015 នៅ 9:54 ល្ងាច • ឆ្លើយតប	JoAnna • January 8, 2015 at 9:54 pm • Reply
មនុស្សដែល, ដោយអាយុឬពិការភាព, មិនអាចទៅយកសំបុត្ររបស់ពួកគេនៅក្នុងប្រអប់សំបុត្រជាការlantbrevbärarservice ទៀងទាត់ផងដែរអាចអនុវត្តសម្រាប់សេវាកម្មខាងក្រោម:	people who, because of age or disability, unable to retrieve their mail in the mailbox as regular lantbrevbärarservice can also apply for the following services:
/កំណត់ហេតុផ្សេងៗ/IDFP/IDFP (615250-02-7): បែបផែន, កិតើ, ពិនិត្យ	/Blog/IDFP/IDFP (615250-02-7): Effects, Dosage, Reviews
A Quiet Place (ភាសាអង់គ្លេស, ភាសាខ្មែរ)	A Quiet Place (English, Khmer)
KMSPICO ប្រព័ន្ធប្រតិបត្តិការ Windows 10 - ទាញយកសកម្មគាំទ្រដោយកតគិតថ្លៃ - Softkelo - រកឃើញកម្មវិធីដែលគ្មានដែនកំណត់, ការបង្ក្រាប & ការ Hack	KMSPICO Windows 10 - Free Download Pro Activator - Softkelo - Find Unlimited Softwares, Cracks & Hacks0

Table 6: Examples of relatively good sentence pairs from the noisy corpus for Khmer–English. Note the lack of word segmentation in Khmer leads to very long tokens.

just the LASER scores that was provided to participants at the outset.

5.2 Methods used by Participants

This year, participants in general used a broader range of features and more sophisticated classifier approaches than previously. We first provide an overview of methods and then give a short summary of each submission.

5.2.1 Methods

Pre-filtering Almost all participants employ pre-filtering rules, based on the length of sentences in terms of tokens or characters, ratio of the lengths, ratio of alpha-numerical tokens, overlap between the English and the foreign sentence (to

	Pashto		Khmer	
	Sentence Pairs	**English Words**	**Sentence Pairs**	**English Words**
dev	3,162	55,439	2,378	40,436
dev test	2,698	46,175	2,309	44,471
test	2,719	47,695	2,320	40,341

Table 7: Statistics for the `flores` test sets used to evaluate the machine translation systems trained on the subsampled data sets. Word counts are obtained with `wc` on tokenized text.

Short Name	**Participant and System Description Citation**
AFRL	Air Force Research Lab, USA
Alibaba	Alibaba, China (Lu et al., 2020)
Bytedance	Bytedance, China (Xu et al., 2020)
Edinburgh	University of Edinburgh, Scotland
Huawei	Huawei, Turkey/China (Açarçiçek et al., 2020)
JHU-Kejriwal	Ankur Kejriwal, Johns Hopkins University, USA (Kejriwal and Koehn, 2020)
JHU-Koerner	Felicia Koerner, Johns Hopkins University, USA (Koerner and Koehn, 2020)
Microsoft	Microsoft, Egypt Development Center, Egypt (Nokrashy et al., 2020)
NRC	National Research Council, Canada (Lo and Joanis, 2020)
UA-Prompsit	University of Alicante and Prompsit, Spain (Esplà-Gomis et al., 2020)
LASER	Officially provided baseline

Table 8: Participants in the shared task.

avoid copy noise), or mismatched email addresses, URLs or numbers.

A common pre-filtering method is also language ID. However mixed results were reported and some participants decided to not use it for Pashto (Açarçiçek et al., 2020).

Some participants worked on morphological segmentation of Khmer but this did not lead to any improvements (Esplà-Gomis et al., 2020; Koerner and Koehn, 2020).

LASER We provided LASER scores that performed well in previous year's filtering task. LASER sentence embeddings are trained as a bottleneck feature for a neural machine translation model and trained on a large collection of parallel corpora in 93 languages[8] which include Khmer but not Pashto. A similarity score for a sentence pair is computed as the cosine distance between the English sentence embedding and the foreign sentence embedding (Nokrashy et al., 2020; Kejriwal and Koehn, 2020; Koerner and Koehn, 2020).

[8] https://github.com/facebookresearch/LASER#supported-languages

Dual cross entropy Neural machine translation systems trained on the provided clean parallel data can be used by feeding in the English sentence and computing the probability of the foreign sentence according to the model, and vice versa. Junczys-Dowmunt (2018) proposed a metric that uses not only the individual computed cross entropy scores but also the difference between them (Lu et al., 2020; Koerner and Koehn, 2020).

Language models To assess the quality of sentences by themselves, i.e., preferring sentences that are fluent in the language, statistical or neural language models are trained, typically using provided Wikipedia and CommonCrawl corpora (Lu et al., 2020; Esplà-Gomis et al., 2020; Kejriwal and Koehn, 2020; Koerner and Koehn, 2020; Lo and Joanis, 2020).

Statistical word translation scores Words in the two sentences should be translation of each other. To what degree this is the case can be assessed with classic word translation models which are learned with the EM algorithm over the clean parallel data (Lu et al., 2020; Lo and Joanis, 2020; Esplà-Gomis et al., 2020).

Classifier An increasing number of participants framed the quality estimation problem as a classification task. This requires positive examples drawn from the provided clean parallel text and negative examples created by corrupting these examples. Typically this involves mismatched sentences, truncated sentences, sentences with swapped word order (Esplà-Gomis et al., 2020; Açarçiçek et al., 2020; Nokrashy et al., 2020; Xu et al., 2020). To create harder negative examples for the classifier, a sentence is paired not with a random sentence from the foreign corpus but with a neighboring sentence of the correctly paired sentence and sentences that have 60% similarity (measured by fuzzy match score) to the correct translation (Açarçiçek et al., 2020).

5.2.2 Individual Submissions

AFRL use their corpus-building method (Erdmann and Gwinnup, 2019) but with a bidirectional quality metric that nearly eliminates pre-filtering (used only for the limit on training line length). The coverage metric encourages the addition of a sentence that improves corpus-level bilingual vocabulary frequencies. The new quality metric is the average of sentence-level NMT scores ("log-likelihoods") in both directions.

Alibaba (Lu et al., 2020) use a number of features that are combined linearly: a bilingual GPT-2 model trained on source-target language pairs as well as monolingual GPT-2 model each of the languages, dual cross entropy from neural machine translation models trained in both directions and statistical word translation model scores. They report that they experimented with classifiers to weight features but found this to be not beneficial.

Bytedance (Xu et al., 2020) tackle only the combined alignment/filtering task. The sentence alignment methods draws on statistical lexical translation scores, as used in YiSi-2. They iteratively improve the lexical model by adding high-quality mined sentence pair to its training data. Their filtering method is a classifier based on monolingual language models and a cross-lingual language model (XLM), followed by an added convolutional layer. They also use language ID and n-gram coverage during a re-ranking stage and ensemble model variations (different architectures, hyper parameters).

Huawei (Açarçiçek et al., 2020) focus on an end-to-end classifier approach that learns to distinguish clean parallel data from misaligned sentence pairs. The model first uses a Transformer model to obtain sentence representations, followed either by a classifier (Siamese network) or additional layers that are fine-tuned. They report better performance with a RoBERTa-style Transformer setup over a BERT-style Transformer. A relatively small training corpus is used (2,000 or 10,000 sentence pairs) with 10x over-sampled negatives.

JHU-Kejriwal (Kejriwal and Koehn, 2020) use LASER scores with some novel transformation of score ranges, language ID confidence scores, monolingual language models trained on words and characters, and length-based filters.

JHU-Koerner (Koerner and Koehn, 2020) employ a linear combination of LASER scores, monolingual language model scores, dual cross entropy, and use a sentence duplication penalty.

Microsoft (Nokrashy et al., 2020) focus on the LASER scores, using both the provided LASER scores, custom LASER scores using a model trained on the provided clean parallel data (which are better for Pashto but worse for Khmer), and a classifier built on a pair of LASER sentence embeddings trained to distinguish between clean sentence pairs and artificially bad sentence pairs. While these three scores fare differently for the two languages pairs, a combination of them performs best.

NRC (Lo and Joanis, 2020) tackle both filtering and alignment. Their filtering score is mainly based on Yisi-2 (Lo, 2019), a language model trained on the target side, and representations obtained with XLM-RoBERTa (Conneau et al., 2020) pre-trained for Pashto, Khmer, and English. Sentence alignment is based on the approach by Moore (2002), first applied to align paragraphs and then sentences.

UA-Prompsit (Esplà-Gomis et al., 2020) use an extended version of the established Bicleaner tool which is a classifier that uses several features ranging from coarse (e.g., statistical word translation models scores) to shallow (e.g., average token length, length ratio, punctuation count). The classifier uses the extremely randomized tree algorithm. They also use a 7-gram character language model as refinement.

LASER scores were provided to participants, with filtering for language ID and maximum 60% overlap between source and target sentence.

Edinburgh did not submit a system description paper.

5.3 Subset Selection

We provided to the participants a file containing one sentence pair per line (see Section 4.3) each for the two languages. A submission to the shared task consists of a file with the same number of lines, with one score per line corresponding to the quality of the corresponding sentence pair.

To evaluate a submitted score file, we selected subsets of a predefined size, defined by the number of English words (5 million). We chose the number of English words instead of Pashto or Khmer words, since the latter would allow selection of sentence pairs with very few non-English words and many English words which are beneficial for decoder training but do not count much towards the non-English word total.

Selecting a subset of sentence pairs is done by finding a threshold score, so that the sentence pairs that will be included in the subset have a quality score at and above this threshold. In some cases, a submission assigned this threshold score to a large number of sentence pairs. Including all of them would yield too large a subset, excluding them yields too small a subset. Hence, we randomly included some of the sentence pairs with the exact threshold score to get the desired size in this case.

5.4 Evaluation System Training

Given a selected subset of a given size for a system submission, we built neural machine translation systems from scratch (SCRATCH) and by continued training on a pre-trained model (MBART) to evaluate the quality of the selected sentence pairs.

SCRATCH For from-scratch training, we used the `fairseq` (Ott et al., 2019) transformer model with the parameter settings shown in Figure 1. Preprocessing was done with sentence piece for a 5000 subword vocabulary on tokenized text using the Moses tokenizer (but no truecasing was used). Decoding was done with beam size 5 and length normalization 1.2. Training a system for the 5 million subsets took about 13 hours, on a single GTX 1080ti GPU. Scores on the test sets were computed with Sacrebleu (Post, 2018). We report case-insensitive scores.

```
--arch transformer
--share-all-embeddings
--encoder-layers 5
--decoder-layers 5
--encoder-embed-dim 512
--decoder-embed-dim 512
--encoder-ffn-embed-dim 2048
--decoder-ffn-embed-dim 2048
--encoder-attention-heads 2
--decoder-attention-heads 2
--encoder-normalize-before
--decoder-normalize-before
--dropout 0.4
--attention-dropout 0.2
--relu-dropout 0.2
--weight-decay 0.0001
--label-smoothing 0.2
--criterion label_smoothed_cross_entropy
--optimizer adam
--adam-betas '(0.9, 0.98)'
--clip-norm 0
--lr-scheduler inverse_sqrt
--warmup-update 4000
--warmup-init-lr 1e-7
--lr 1e-3 --min-lr 1e-9
--max-tokens 4000
--update-freq 4
--max-epoch 100
--save-interval 10
```

Figure 1: The baseline `flores` model settings[9] for the NMT training from scratch with `fairseq`

MBART For mBART evaluation, we initialize the weights of transformer with the mBART bilingual pre-training. We used monolingual text from CommonCrawl with denoising objective to pre-train the transformer. We trained 2 bilingual mBART models, one with English and Pashto text and another with English and Khmer text. Both these models were pre-trained with batch size of 256 for 500,000 updates, which took about 57 hours on 16 V100 GPUs.

Continued training on the filtered subsets uses some different parameter settings, as listed in Figure 2. This continued training is faster; it takes about half as much time.

6 Results

In this section we present the results of the shared task evaluation. We added additional unofficial condition at 2, 3, and 7 million English words, to better observe tendencies.

6.1 Core Results

The results are reported in Table 9 (Pashto) and Table 10 (Khmer). The tables contains the BLEU

[9] https://github.com/facebookresearch/flores#train-a-baseline-transformer-model

```
--dropout 0.1
--attention-dropout 0.1
--relu-dropout 0.0
--weight-decay 0.0
--label-smoothing 0.1
--adam-eps 1e-06
--lr 0.0001
--max-update 100000
--patience 10
```

Figure 2: Different model settings for continued training of the provided mBART model. The other settings are the same.

scores for

- development test set and final test set
- neural machine translation from scratch and mBART pre-training
- 2, 3, 5 and 7 million word subsets.

The official scoring is for the 5 million word data settings on the final test set. In the table, we highlight cells for the best scores for each of these settings, as well as scores that are close to it. Results for the unofficial 2, 3 and 7 million word baseline are shown without highlighting.

For almost all submission the highest BLEU scores is reached with subsets of 5 million words. There is also fairly high consistency between relative performance under training from scratch and mBART training. The best showings are by Alibaba and Huawei, followed by NRC and UA-Prompsit, with Microsoft still competitive. Other submissions score at least 1 BLEU points behind these.

Participants that also worked on sentence alignment of the provided document pairs were able to outperform the provided sentence pairs. The peak for these submissions shifts in most cases to the 7 million word subset. So, they were able to extract more useful sentence pairs. The best submissions for this setup comes from Bytedance. They outperform the provided sentence pairs and LASER scores by +3.8 BLEU (from 7.7 to 11.5) for Pashto from-scratch, +2.6 BLEU (from 10.3 to 12.9) for Pashto mBART, +4.3 BLEU (from 8.4 to 12.7) for Khmer from-scratch, +2.6 BLEU (from 12.9 to 15.5) for Khmer mBART.

6.2 Variance in the Evaluation

During the exploration of the evaluation protocol, we had some concerns about the stability of the BLEU scores obtained from training runs on a data set. This concern was reinforced by feedback from participants who did not match the baseline scores that we reported on the shared task web page.

To assess this, we executed three training runs for each subset of 5 million words selected from participant submissions. The resulting scores vary at most by 0.3 BLEU points for an identical training corpus, and differ most frequently just 0.1 BLEU point difference or are identical across all runs. The official reported results in Tables 9 and 10 are the average score across these three runs.

There may be higher differences for training on different hardware. We used a single NVidia GeForce GTX 1080ti GPU.

6.3 Average Sentence Length

Given the quality scores, subsets are selected by including the highest ranked sentence pairs until the total number of English words in these sentences reaches the specified size. So, if a quality scores prefers shorter sentences, more sentences are selected. It is not clear in general, all things being otherwise equal, if shorter or longer sentences are better for training machine translation systems.

What choices did the participants make in their quality scores? Table 11 and Table 12 show the number of sentences and the corresponding average number of words per sentence for the official subsets for all submissions. The average sentence length differs quite significantly, ranging from 12.3 to 29.0 words per sentence for Pashto, and 17.0 to 27.3 words per sentence for Khmer. Cross-referencing this against the effectiveness of the scores, methods that selected shorter sentences on average performed better.

In contrast to this, the average sentence length of submissions that also tackled sentence alignment is longer when compared to each participant's filtering-only submission.

6.4 Diversity of Submissions

The different submissions subselect different sentences, but how different are they?

Tables 13 and 14 give detailed statistics about how many sentence pairs the subsets of any two submissions for the two languages and two data conditions have in common.

The tables show for the 5 million word subset selected for each submission how many sentence pairs it contains (e.g., AFRL: 172,145), how many

Pashto	2 million				3 million				5 million				7 million			
	SCRATCH		MBART		SCRATCH		MBART		SCRATCH		MBART		SCRATCH		MBART	
	DEVT	TEST	DEVT	TEST	DEVT	TEST	DEVT	TEST	DEVT	TEST	DEVT	TEST	DEVT	TEST	DEVT	TEST
AFRL	6.2	4.8	9.6	8.7	7.4	5.9	10.7	9.8	9.4	8.2	11.2	10.1	9.3	7.4	11.0	9.1
Alibaba	9.9	8.4	12.0	11.0	10.3	9.4	12.6	11.6	10.8	9.5	13.1	12.2	10.0	8.8	12.8	11.6
Edinburgh	9.6	8.5	11.4	10.8	10.3	8.5	11.6	10.5	10.0	8.3	11.3	10.5	9.6	7.7	11.6	9.7
Huawei	9.7	8.6	11.5	10.6	10.7	9.3	12.3	11.7	10.9	9.7	13.3	12.2	-	-	12.6	10.1
JHU-Kejriwal 0.8-5	8.0	6.7	10.5	9.9	9.1	7.7	10.8	10.1	9.7	7.8	11.3	10.2	9.4	7.2	11.5	10.0
JHU-Kejriwal 0.9-0	7.9	6.7	10.4	9.9	9.1	7.3	11.0	10.5	9.6	8.0	11.7	10.2	9.4	7.9	11.6	10.3
JHU-Kejriwal 0.9-5	8.2	6.9	10.2	9.8	9.1	7.6	11.0	10.2	9.6	7.7	11.6	10.4	9.4	7.5	11.4	9.9
JHU-Koerner dual-xent	7.7	6.0	9.4	9.4	8.9	7.6	11.3	10.7	9.8	8.0	10.9	10.3	9.5	7.4	11.3	9.5
JHU-Koerner laser-lm	9.1	7.7	11.0	10.4	9.7	8.4	11.4	10.3	9.9	8.3	11.1	10.0	9.5	7.8	11.0	9.6
LASER	9.1	7.6	10.9	10.2	9.4	7.8	11.0	10.3	9.7	7.7	11.4	10.3	9.7	8.2	11.1	9.8
Microsoft	9.4	8.5	11.2	10.6	10.5	9.2	11.7	11.1	10.1	8.5	12.8	11.6	9.9	8.5	11.7	10.3
NRC	8.7	7.5	9.3	8.8	10.2	8.6	11.4	10.7	10.5	8.9	12.9	12.0	9.8	8.5	12.5	11.5
UA-Prompsit	9.9	9.2	11.2	10.8	10.3	9.5	11.8	11.1	10.8	9.2	12.6	11.7	10.2	8.4	11.7	10.3
Alibaba alignment	9.1	8.8	11.7	10.9	10.8	10.0	12.2	11.8	11.7	10.4	13.2	12.4	11.2	9.8	12.8	11.8
Bytedance alignment	11.2	9.9	12.1	11.4	11.7	10.7	12.8	12.3	12.2	11.4	13.4	12.8	12.9	11.5	13.6	12.9
NRC alignment	11.4	10.1	12.2	11.1	12.0	10.5	12.7	11.7	11.8	10.5	13.4	12.4	11.1	10.0	13.1	11.9

Table 9: Results for Pashto: BLEU scores are reported for systems trained on 2, 3, 5 and 7 million word subsets of the data, subsampled based on the quality scores provided by the participants.

Khmer	2 million				3 million				5 million				7 million			
	SCRATCH		MBART		SCRATCH		MBART		SCRATCH		MBART		SCRATCH		MBART	
	DEVT	TEST	DEVT	TEST	DEVT	TEST	DEVT	TEST	DEVT	TEST	DEVT	TEST	DEVT	TEST	DEVT	TEST
Alibaba	8.2	9.3	10.3	12.5	8.7	10.3	10.9	12.9	8.9	11.0	11.5	14.0	7.8	10.1	10.6	13.2
Huawei	8.5	9.8	10.2	13.0	8.8	10.5	11.1	13.8	8.8	10.8	11.4	14.0	8.2	10.5	11.1	14.0
JHU Kejriwal 0.8-6	6.6	7.9	9.4	11.3	6.9	8.3	9.7	12.0	7.1	8.3	9.8	12.5	6.7	7.8	10.1	12.1
JHU-Kejriwal 0.8-5-filt	6.4	7.6	9.2	11.4	6.6	7.9	10.1	12.2	7.1	8.4	9.9	12.7	6.3	7.6	10.1	12.2
JHU-Kejriwal 0.8-5	5.5	6.1	6.0	8.1	5.9	6.8	6.8	7.9	6.5	7.4	10.0	12.1	6.5	7.8	9.8	12.2
LASER	6.4	7.7	9.2	10.9	7.0	8.0	9.7	12.0	7.1	8.4	10.5	12.9	6.7	8.6	10.5	12.6
Microsoft	7.2	8.7	9.7	11.9	8.0	9.3	10.3	12.5	7.8	9.3	11.2	13.3	7.8	9.7	11.1	13.7
NRC	7.7	9.5	10.4	12.6	8.5	10.6	10.5	13.4	8.7	10.8	11.2	13.7	8.4	10.3	11.2	13.8
UA-Prompsit	7.9	9.1	10.0	12.2	8.4	9.7	10.7	13.0	8.4	10.0	10.8	13.8	7.6	9.4	10.9	13.2
Bytedance alignment	9.3	11.2	11.2	14.0	9.8	11.8	11.7	14.6	10.5	12.7	12.3	14.9	10.3	12.5	12.7	15.5
NRC alignment	8.3	9.9	10.3	12.6	8.5	10.8	11.0	13.2	9.1	11.3	11.5	14.2	9.4	11.9	11.7	14.5

Table 10: Results for Khmer: BLEU scores are reported for systems trained on 2, 3, 5 and 7 million word subsets of the data, subsampled based on the quality scores provided by the participants.

Pashto	Sentences	Words/S
AFRL	172,145	29.0
Alibaba	375,507	13.3
Edinburgh	274,021	18.2
Huawei	383,554	13.0
JHU Kejriwal 0.8-5	208,922	23.9
JHU Kejriwal 0.9-0	257,060	19.5
JHU Kejriwal 0.9-5	209,059	23.9
JHU-Koerner laser-lm	225,750	22.1
JHU-Koerner dual-xent	205,346	24.3
LASER	225,725	22.2
Microsoft	238,612	21.0
NRC	405,330	12.3
UA-Prompsit	315,133	15.9
Alibaba alignment	222,539	22.5
Bytedance alignment	219,887	22.7
NRC alignment	244,622	20.4

Table 11: Number of sentences and the corresponding average sentence length (counting English words) for Pashto.

Khmer	Sentences	Words/S
Alibaba	258,044	19.4
Huawei	278,534	18.0
JHU Kejriwal 0.8-5	218,851	22.7
JHU Kejriwal 0.8-5-filt	191,864	26.0
JHU Kejriwal 0.8-6	182,126	27.3
LASER	240,978	20.7
Microsoft	256,762	19.4
NRC	293,414	17.0
UA-Prompsit	206,018	24.3
Bytedance alignment	169,492	29.5
NRC alignment	264,796	18.8

Table 12: Number of sentences and the corresponding average sentence length (counting English words) for Khmer.

sentence pairs are unique to this submission's subset (e.g., AFRL: 7.6% of the 172,145 sentence pairs) and how many are in common with other submission (e.g,, 59.2% of AFRL's subset are also in Alibaba's subset).

The leading submissions show mostly about 60% overlap, although there are also more similar submissons (Alibaba's and Huawai's share around 80% of sentence pairs). The alignment submissions tend to be quite different, not surprisingly.

7 Conclusion

We report on the findings of the WMT 2020 Shared Task on Parallel Corpus Filtering and Alignment. Ten participants used a variety of methods that gave quite different results, as measured by translation quality, optimal subset sizes, sentence length, etc. We hope that this task provides a benchmark for future research and improvements on this task.

References

Roee Aharoni, Melvin Johnson, and Orhan Firat. 2019. Massively multilingual neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3874–3884, Minneapolis, Minnesota. Association for Computational Linguistics.

Alexandra Antonova and Alexey Misyurev. 2011. Building a web-based parallel corpus and filtering out machine-translated text. In *Proceedings of the 4th Workshop on Building and Using Comparable Corpora: Comparable Corpora and the Web*, pages 136–144, Portland, Oregon. Association for Computational Linguistics.

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. 2011. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 355–362, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Haluk Açarçiçek, Talha Çolakoğlu, pınar ece aktan hatipoğlu, Chong Hsuan Huang, and Wei Peng. 2020. Filtering noisy parallel corpus using transformers with proxy task learning. In *Proceedings of the Fifth Conference on Machine Translation (WMT)*.

D. Bahdanau, K. Cho, and Y. Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations (ICLR)*.

Submission	Total	Unique	AFRL	Alibaba	Alibaba alignment	Bytedance alignment	Edinburgh	Huawei	JHU-Kejriwal 0.8-5	JHU-Kejriwal 0.9-0	JHU-Kejriwal 0.9-5	JHU-Koerner dual-xent	JHU-Koerner laser-lm	LASER	Microsoft	NRC	NRC alignment	UA-Prompsit
AFRL	172145	7.6%	-	59.2%	6.0%	19.2%	61.3%	57.2%	42.6%	43.3%	42.7%	53.8%	47.5%	47.5%	51.6%	60.5%	30.9%	60.2%
Alibaba	375507	4.6%	27.1%	-	3.9%	12.5%	45.9%	82.2%	36.6%	44.6%	36.8%	41.6%	36.8%	36.8%	42.8%	66.4%	31.1%	59.2%
Alibaba alignment	222539	62.8%	4.6%	6.6%	-	32.4%	5.9%	6.6%	3.5%	3.4%	3.5%	5.2%	3.9%	3.9%	5.1%	6.5%	7.9%	6.0%
Bytedance alignment	219887	45.2%	15.1%	21.3%	32.8%	-	19.1%	22.0%	14.0%	13.6%	14.1%	18.1%	15.0%	15.0%	18.1%	21.1%	19.4%	19.2%
Edinburgh	274021	5.6%	38.5%	62.9%	4.8%	15.4%	-	60.3%	50.4%	51.9%	50.6%	46.3%	58.7%	58.7%	53.4%	56.3%	32.4%	59.9%
Huawei	383554	3.0%	25.7%	80.5%	3.9%	12.6%	43.1%	-	36.4%	44.0%	36.6%	42.3%	34.1%	34.1%	42.1%	72.3%	31.2%	59.2%
JHU-Kejriwal 0.8-5	208922	0.2%	35.1%	65.8%	3.7%	14.7%	66.1%	66.8%	-	95.7%	98.5%	63.8%	74.8%	74.8%	64.8%	57.6%	31.1%	58.6%
JHU-Kejriwal 0.9-0	257060	2.0%	29.0%	65.2%	3.0%	11.7%	55.4%	65.6%	77.7%	-	78.0%	56.5%	68.3%	68.3%	59.0%	55.2%	26.4%	54.0%
JHU-Kejriwal 0.9-5	209059	0.0%	35.2%	66.2%	3.7%	14.8%	66.3%	67.2%	98.5%	95.9%	-	63.9%	75.1%	75.1%	65.1%	58.0%	31.4%	58.9%
JHU-Koerner dual-xent	205346	1.6%	45.1%	76.0%	5.6%	19.4%	61.8%	79.1%	64.9%	70.7%	65.1%	-	56.6%	56.6%	62.3%	68.5%	39.0%	68.1%
JHU-Koerner laser-lm	225750	0.0%	36.3%	61.2%	3.9%	14.6%	71.3%	58.0%	69.2%	77.8%	69.5%	51.5%	-	100.0%	77.3%	50.8%	28.5%	54.8%
LASER	225725	0.0%	36.3%	61.2%	3.9%	14.6%	71.3%	58.0%	69.2%	77.8%	69.5%	51.5%	100.0%	-	77.3%	50.8%	28.5%	54.8%
Microsoft	238612	0.9%	37.2%	67.4%	4.7%	16.7%	61.3%	67.7%	56.7%	63.5%	57.0%	53.6%	73.1%	73.1%	-	66.6%	32.5%	58.5%
NRC	405330	12.7%	25.7%	61.5%	3.6%	11.4%	38.1%	68.4%	29.7%	35.0%	29.9%	34.7%	28.3%	28.3%	39.2%	-	29.7%	50.1%
NRC alignment	244622	42.2%	21.7%	47.8%	7.2%	17.5%	36.3%	48.9%	26.6%	27.7%	26.8%	32.8%	26.3%	26.3%	31.7%	49.1%	-	41.7%
UA-Prompsit	315133	7.4%	32.9%	70.6%	4.3%	13.4%	52.1%	72.0%	38.8%	44.1%	39.1%	44.4%	39.2%	39.2%	44.3%	64.5%	32.4%	-

Table 13: **Overlap for Pashto.** For each submission, a row in the table lists the total number of sentence pairs, the ratio of unique sentence pairs that are in included in no other submission, and the ratio of sentence pairs shared with each of the other submissions.

Submission	Total	Unique	Alibaba	Bytedance	Huawei	JHU Kejriwal 0.8-6	JHU Kejriwal 0.8-5	JHU Kejriwal 0.8-5-filt	LASER	Microsoft	NRC	NRC alignment	UA-Prompsit
Alibaba	258044	13.7%	-	19.1%	68.7%	35.7%	35.3%	37.1%	41.9%	54.8%	59.6%	32.7%	41.1%
Bytedance alignment	169492	62.6%	29.0%	-	29.6%	18.7%	15.9%	18.8%	19.4%	24.9%	27.1%	24.0%	21.0%
Huawei	278534	11.4%	63.7%	18.0%	-	33.7%	38.1%	35.2%	41.8%	53.5%	58.1%	30.7%	42.4%
JHU Kejriwal 0.8-6	182126	0.2%	50.6%	17.4%	51.5%	-	78.0%	99.0%	91.1%	82.6%	47.2%	26.0%	31.3%
JHU-Kejriwal 0.8-5	218851	11.9%	41.7%	12.3%	48.5%	64.9%	-	68.8%	66.3%	60.8%	43.9%	20.3%	24.4%
JHU-Kejriwal 0.8-5-filt	191864	0.1%	49.9%	16.6%	51.1%	94.0%	78.5%	-	91.6%	82.8%	46.4%	25.3%	30.7%
LASER	240978	6.2%	44.8%	13.6%	48.3%	68.8%	60.2%	72.9%	-	82.1%	42.3%	22.4%	28.7%
Microsoft	256762	4.4%	55.1%	16.4%	58.0%	58.6%	51.9%	61.8%	77.0%	-	48.7%	26.9%	33.8%
NRC	293414	26.1%	52.4%	15.7%	55.1%	29.3%	32.8%	30.3%	34.8%	42.6%	-	32.1%	34.5%
NRC alignment	264796	58.9%	31.9%	15.3%	32.3%	17.9%	16.8%	18.4%	20.4%	26.1%	35.6%	-	21.6%
UA-Prompsit	206018	28.1%	51.5%	17.3%	57.4%	27.7%	25.9%	28.6%	33.6%	42.1%	49.2%	27.8%	-

Table 14: **Overlap for Khmer.** For each submission, a row in the table lists the total number of sentence pairs, the ratio of unique sentence pairs that are in included in no other submission, and the ratio of sentence pairs shared with each of the other submissions.

Marta Bañón, Pinzhen Chen, Barry Haddow, Kenneth Heafield, Hieu Hoang, Miquel Esplà-Gomis, Mikel L. Forcada, Amir Kamran, Faheem Kirefu, Philipp Koehn, Sergio Ortiz Rojas, Leopoldo Pla Sempere, Gema Ramírez-Sánchez, Elsa Sarrías, Marek Strelec, Brian Thompson, William Waites, Dion Wiggins, and Jaume Zaragoza. 2020. ParaCrawl: Web-scale acquisition of parallel corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4555–4567, Online. Association for Computational Linguistics.

Yonatan Belinkov and Yonatan Bisk. 2017. Synthetic and natural noise both break neural machine translation. *CoRR*, abs/1711.02173.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, et al. 2017. Findings of the 2017 conference on machine translation (wmt17). In *Proceedings of the Second Conference on Machine Translation*, pages 169–214.

Fabienne Braune and Alexander Fraser. 2010. Improved unsupervised sentence alignment for symmetrical and asymmetrical parallel corpora. In *Coling 2010: Posters*, pages 81–89, Beijing, China. Coling 2010 Organizing Committee.

Christian Buck and Philipp Koehn. 2016. Findings of the wmt 2016 bilingual document alignment shared task. In *Proceedings of the First Conference on Machine Translation*, pages 554–563, Berlin, Germany. Association for Computational Linguistics.

Marine Carpuat, Yogarshi Vyas, and Xing Niu. 2017. Detecting cross-lingual semantic divergence for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 69–79, Vancouver. Association for Computational Linguistics.

Vishrav Chaudhary, Yuqing Tang, Francisco Guzmán, Holger Schwenk, and Philipp Koehn. 2019. Low-resource corpus filtering using multilingual sentence embeddings. In *Proceedings of the Fourth Conference on Machine Translation (WMT)*.

Stanley F. Chen. 1993. Aligning sentences in bilingual corpora using lexical information. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics (ACL)*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451.

Lei Cui, Dongdong Zhang, Shujie Liu, Mu Li, and Ming Zhou. 2013. Bilingual data cleaning for SMT using graph-based random walk. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 340–345, Sofia, Bulgaria. Association for Computational Linguistics.

Ahmed El-Kishky, Ahmed Chaudhary, Francisco Guzman, and Philipp Koehn. 2020a. Ccaligned: A massive collection of cross-lingual web-document pairs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020)*.

Ahmed El-Kishky and Francisco Guzmán. 2020. Massively multilingual document alignment with cross-lingual sentence-mover's distance. *arXiv preprint arXiv:2002.00761*.

Ahmed El-Kishky, Philipp Koehn, and Holger Schwenk. 2020b. Searching the web for cross-lingual parallel data. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2417–2420.

Grant Erdmann and Jeremy Gwinnup. 2019. Quality and coverage: The afrl submission to the wmt19 parallel corpus filtering for low-resource conditions task. In *Proceedings of the Fourth Conference on Machine Translation (WMT)*.

Miquel Esplà-Gomis, Víctor M. Sánchez-Cartagena, Jaume Zaragoza-Bernabeu, and Felipe Sánchez-Martínez. 2020. Bicleaner at WMT 2020: Universitat d'alacant–prompsit's submission to the parallel corpus filtering shared task. In *Proceedings of the Fifth Conference on Machine Translation (WMT)*.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, et al. 2020. Beyond english-centric multilingual machine translation. *arXiv preprint arXiv:2010.11125*.

William A. Gale and Kenneth Ward Church. 1993. A program for aligning sentences in bilingual corpora. *Computational Linguistics*, 19(1).

Jonas Gehring, Michael Auli, David Grangier, and Yann N Dauphin. 2016. A convolutional encoder model for neural machine translation. *arXiv preprint arXiv:1611.02344*.

Luís Gomes and Gabriel Pereira Lopes. 2016. First steps towards coverage-based document alignment. In *Proceedings of the First Conference on Machine Translation*, pages 697–702, Berlin, Germany. Association for Computational Linguistics.

Francisco Guzmán, Peng-Jen Chen, Myle Ott, Juan Pino, Guillaume Lample, Philipp Koehn, Vishrav Chaudhary, and Marc'Aurelio Ranzato. 2019. Two new evaluation datasets for low-resource machine

translation: Nepali-english and sinhala-english. *arXiv preprint arXiv:1902.01382*.

Vu Cong Duy Hoang, Philipp Koehn, Gholamreza Haffari, and Trevor Cohn. 2018. Iterative back-translation for neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 18–24. Association for Computational Linguistics.

Marcin Junczys-Dowmunt. 2018. Dual conditional cross-entropy filtering of noisy parallel corpora. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 888–895, Belgium, Brussels. Association for Computational Linguistics.

Ankur Kejriwal and Philipp Koehn. 2020. An exploratory approach to the parallel corpus filtering shared task WMT20. In *Proceedings of the Fifth Conference on Machine Translation (WMT)*.

Huda Khayrallah and Philipp Koehn. 2018. On the impact of various types of noise on neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 74–83. Association for Computational Linguistics.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of the Tenth Machine Translation Summit (MT Summit X)*, Phuket, Thailand.

Philipp Koehn, Francisco Guzmán, Vishrav Chaudhary, and Juan Pino. 2019. Findings of the WMT 2019 shared task on parallel corpus filtering for low-resource conditions. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 54–72, Florence, Italy. Association for Computational Linguistics.

Philipp Koehn, Huda Khayrallah, Kenneth Heafield, and Mikel L. Forcada. 2018. Findings of the wmt 2018 shared task on parallel corpus filtering. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 726–739, Belgium, Brussels. Association for Computational Linguistics.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39, Vancouver. Association for Computational Linguistics.

Felicia Koerner and Philipp Koehn. 2020. Dual conditional cross entropy scores and laser similarity scores for the WMT20 parallel corpus filtering shared task. In *Proceedings of the Fifth Conference on Machine Translation (WMT)*.

Gaurav Kumar, George Foster, Colin Cherry, and Maxim Krikun. 2019. Reinforcement learning based curriculum optimization for neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2054–2061, Minneapolis, Minnesota. Association for Computational Linguistics.

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Phrase-based & neural unsupervised machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5039–5049, Brussels, Belgium. Association for Computational Linguistics.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation.

Chi-kiu Lo. 2019. YiSi - a unified semantic MT quality evaluation and estimation metric for languages with different levels of available resources. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 507–513, Florence, Italy. Association for Computational Linguistics.

Chi-kiu Lo and Eric Joanis. 2020. Iteratively refined statistical sentence alignment and improved bilingual mappings of pretrained multilingual language model for identifying bette rparallel MT training data. In *Proceedings of the Fifth Conference on Machine Translation (WMT)*.

Jun Lu, Xin Ge, Yangbin Shi, and Yuqi Zhang. 2020. Alibaba submission to the WMT20 parallel corpus filtering task. In *Proceedings of the Fifth Conference on Machine Translation (WMT)*.

Robert C. Moore. 2002. Fast and accurate sentence alignment of bilingual corpora. In *Machine Translation: From Research to Real Users, 5th Conference of the Association for Machine Translation in the Americas, AMTA 2002 Tiburon, CA, USA, October 6-12, 2002, Proceedings*, volume 2499 of *Lecture Notes in Computer Science*. Springer.

Muhammad El Nokrashy, Amr Hendy, Mohamed Abdelghaffar, Mohamed Afify, Ahmed Tawfik, and Hany Hassan Awadalla. 2020. Score combination for improved parallel corpus filtering for low resource conditions. In *Proceedings of the Fifth Conference on Machine Translation (WMT)*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting bleu scores. In *Proceedings of the Third Conference on*

Machine Translation: Research Papers, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

Alexandre Rafalovitch and Robert Dale. 2009. United Nations General Assembly resolutions: A six-language parallel corpus. In *Proceedings of the Twelfth Machine Translation Summit (MT Summit XII)*. International Association for Machine Translation.

Spencer Rarrick, Chris Quirk, and Will Lewis. 2011. MT detection in web-scraped parallel corpora. In *Proceedings of the 13th Machine Translation Summit (MT Summit XIII)*, pages 422–430. International Association for Machine Translation.

Philip Resnik. 1999. Mining the web for bilingual text. In *Proceedings of the 37th Annual Meeting of the Association of Computational Linguistics (ACL)*.

Holger Schwenk. 2018. Filtering and mining parallel data in a joint multilingual space. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 228–234. Association for Computational Linguistics.

Rico Sennrich and Martin Volk. 2010. MT-based sentence alignment for OCR-generated parallel texts. In *Proceedings of the Ninth Conference of the Association for Machine Translation in the Americas*.

Jason R Smith, Herve Saint-Amand, Magdalena Plamada, Philipp Koehn, Chris Callison-Burch, and Adam Lopez. 2013. Dirt cheap web-scale parallel text from the common crawl. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1374–1383.

Wolfgang Täger. 2011. The sentence-aligned european patent corpus. In *Proceedings of the 15th International Conference of the European Association for Machine Translation (EAMT)*, pages 177–184.

Kaveh Taghipour, Shahram Khadivi, and Jia Xu. 2011. Parallel corpus refinement as an outlier detection algorithm. In *Proceedings of the 13th Machine Translation Summit (MT Summit XIII)*, pages 414–421. International Association for Machine Translation.

Brian Thompson and Philipp Koehn. 2019. Vecalign: Improved sentence alignment in linear time and space. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1342–1348, Hong Kong, China. Association for Computational Linguistics.

Brian Thompson and Philipp Koehn. 2020. Exploiting sentence order in document alignment. *arXiv preprint arXiv:2004.14523*.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, pages 2214–2218, Istanbul, Turkey. European Language Resources Association (ELRA). ACL Anthology Identifier: L12-1246.

Dániel Varga, Péter Halaácsy, András Kornai, Viktor Nagy, László Németh, and Viktor Trón. 2005. Parallel corpora for medium density languages. In *Proceedings of the RANLP 2005 Conference*, pages 590–596.

Dániel Varga, Péter Halácsy, András Kornai, Viktor Nagy, László Németh, and Viktor Trón. 2007. Parallel corpora for medium density languages. *Amsterdam Studies In The Theory And History Of Linguistic Science Series 4*, 292:247.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Ashish Venugopal, Jakob Uszkoreit, David Talbot, Franz Och, and Juri Ganitkevitch. 2011. Watermarking the outputs of structured prediction with an application in statistical machine translation. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1363–1372, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Wei Wang, Taro Watanabe, Macduff Hughes, Tetsuji Nakagawa, and Ciprian Chelba. 2018. Denoising neural machine translation training with trusted data and online data selection. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 133–143, Belgium, Brussels. Association for Computational Linguistics.

Marlies van der Wees, Arianna Bisazza, and Christof Monz. 2017. Dynamic data selection for neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1411–1421. Association for Computational Linguistics.

Hainan Xu and Philipp Koehn. 2017. Zipporah: a fast and scalable data cleaning system for noisy web-crawled parallel corpora. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2935–2940. Association for Computational Linguistics.

Runxin Xu, Zhuo Zhi, Jun Cao, Mingxuan Wang, and Lei Li. 2020. Volctrans parallel corpus filtering system for WMT 2020. In *Proceedings of the Fifth Conference on Machine Translation (WMT)*.

Michał Ziemski, Marcin Junczys-Dowmunt, and Bruno Pouliquen. 2015. The united nations parallel corpus v1.0. In *International Conference on Language Resources and Evaluation (LREC)*.

Findings of the WMT 2020 Shared Task on Quality Estimation

Lucia Specia,[1,2] Frédéric Blain,[2,3] Marina Fomicheva,[2] Erick Fonseca,[4]
Vishrav Chaudhary,[5] Francisco Guzmán,[5] André F. T. Martins[4,6]

[1]Imperial College London, [2]University of Sheffield, [3]University of Wolverhampton,
[4]Instituto de Telecomunicações, [5]Facebook AI, [6]Unbabel
{l.specia,m.fomicheva}@sheffield.ac.uk, f.blain@wlv.ac.uk
erick.fonseca@lx.it.pt, {vishrav,fguzman}@fb.com,
andre.martins@unbabel.com

Abstract

We report the results of the WMT20 shared task on Quality Estimation, where the challenge is to predict the quality of the output of neural machine translation systems at the word, sentence and document levels. This edition included new data with open domain texts, direct assessment annotations, and multiple language pairs: English–German, English–Chinese, Russian–English, Romanian–English, Estonian–English, Sinhala–English and Nepali–English data for the sentence-level subtasks, English–German and English–Chinese for the word-level subtask, and English–French data for the document-level subtask. In addition, we made neural machine translation models available to participants. 19 participating teams from 27 institutions submitted altogether 1374 systems to different task variants and language pairs.

1 Introduction

This shared task builds on its previous eight editions to further examine automatic methods for estimating the quality of neural machine translation (MT) output at run-time, without the use of reference translations. As in previous editions, it includes the (sub)tasks of word-level, sentence-level and document-level estimation. Important elements introduced this year are: a variant of the sentence-level task where sentences are annotated with *direct assessment* (DA)[1] scores instead of labels based on post-editing; a new multilingual sentence-level dataset mainly from Wikipedia articles, where the source articles can be retrieved for document-wide context; the availability of NMT models to explore system-internal information for the task.

In addition to advancing the state of the art at all prediction levels, our main goals are:

- To create a new set of public benchmarks for tasks in quality estimation.
- To investigate models for predicting DA scores and their relationship with models trained for predicting post-editing effort,
- To study the feasibility of multilingual (or even language independent) approaches to QE.
- To study the influence of source-language document-level context for the task of QE.
- To analyse the applicability of NMT model information for QE.

We have three subtasks: Task 1 aims at predicting DA scores at sentence level (Section 2.1); Task 2 aims at predicting post-editing effort scores at both sentence and word levels, i.e. words that need editing, as well as missing words and incorrect source words (Section 2.2); Task 3 aims at predicting a score for an entire document as a function of the proportion of incorrect words in such a document, weighted by the severity of the different errors (Section 2.3).

Tasks make use of large datasets produced from either post-editions or DA annotations, or error annotation, all done by professional translators. The text domains vary for each subtask. Neural MT systems were built on freely available data using an open-source toolkit to produce translations, and these models were made available to participants. We provide new training and test datasets for Tasks 1 and 2, and a new test set for Task 3. The datasets

[1]We note that the procedure followed for our data diverges from that proposed by Graham et al. (2016) in three ways: (a) we employ fewer but professional translators to score each sentence, (b) scoring is done against the source segment (bilingual annotation) and not the reference, and (c) we provide translators with guidelines on the meaning of ranges of scores.

Proceedings of the 5th Conference on Machine Translation (WMT), pages 743–764

and models released are publicly available. Participants are also allowed to explore any additional data and resources deemed relevant.

Baseline systems were entered in the platform by the task organisers (Section 3). The shared task uses CodaLab as submission platform, where participants (Section 4) could submit up to 30 systems for each task and language pair. Results for all tasks evaluated according to standard metrics are given in Section 5, while a discussion on the main goals and findings from this year's task is presented in Section 6.

2 Subtasks

In what follows we give a brief description for each subtask, including the datasets provided for them.

2.1 Task 1: Predicting sentence-level DA

This task consists in scoring translation sentences according to their perceived quality score – which we refer to as direct assessment (DA). For that, a **new dataset**, was created containing seven languages pairs using sentences mostly from Wikipedia[2]. These language pairs are divided into 3 categories: the high-resource English→German (En-De), English→Chinese (En-Zh) and Russian→English (Ru-En) pairs; the medium-resource Romanian→English (Ro-En) and Estonian→English (Et-En) pairs; and the low-resource Sinhala→English (Si-En) and Nepali→English (Ne-En) pairs.

Translations were produced with state-of-the-art transformer-based NMT models trained using publicly available data and the fairseq toolkit (Ott et al., 2019); and were manually annotated for perceived quality. The quality label for this task ranges from 0 to 100, following the FLORES guidelines (Guzmán et al., 2019). According to the guidelines given to annotators, the 0-10 range represents an incorrect translation; 11-29, a translation with few correct keywords, but the overall meaning is different from the source; 30-50, a translation with major mistakes; 51-69, a translation which is understandable and conveys the overall meaning of the source but contains typos or grammatical errors; 70-90, a translation that closely preserves the semantics of the source sentence; and 91-100, a perfect translation.

[2]This dataset is a superset of MLQE (Fomicheva et al., 2020c) which included 6 language pairs and is sourced entirely from Wikipedia. The newly-added English-Russian DAs follow the same guidelines, but come from diverse sources.

Statistics on the dataset are shown in Table 1. More details are given in Fomicheva et al. (2020a). The complete data can be downloaded from the public repository[3].

Participation was encouraged for each language pair and also for the **multilingual variant** of the task, where submissions had to include predictions for all six Wikipedia-based language pairs (all except Ru-En). The latter aimed at fostering work on language-independent models, as well as models that can leverage data from multiple languages.

2.2 Task 2: Predicting post-editing effort

This task follows from previous editions of the WMT shared task and consists in scoring translations according to the proportion of their words that need to be fixed using HTER as label, i.e. the minimum edit distance between the machine translation and its manually post-edited version, as well as detecting where errors are in the translation of source sentences. It uses a subset of the languages from Task 1, namely the two high-resource language pairs (En-De and En-Zh, Table 1).

Sentence-level post-editing effort The label for this task is the percentage of edits that need to be fixed (HTER). Starting with the En-De and En-Zh source-machine translation segment pairs, the machine translation sentences were post-edited by two human translators, one per language, who are paid editors from the Unbabel community. The two translators had no access to the direct assessments above. In other words, the DA and HTER annotations were collected independently.

The average human translation error rate between the machine translated text and the post-edited text was 0.32 for En-De, and 0.62 for En-Zh. HTER labels were computed using TERCOM [4] with default settings (tokenised, case insensitive, exact matching only), with scores capped to 1.

Word-level errors This variant evaluates the extent to which we can detect word-level errors in MT output. Based on the post-edited translations, as described above, we annotate each token of the target and the source sentence, as well as word omission errors. The code to produce this set of tags from any prior WMT corpora is available for

[3]https://github.com/sheffieldnlp/mlqe-pe

[4]https://github.com/jhclark/tercom

	Sentences			Tokens				
Languages	Train	Dev	Test	Train	Dev	Test	DA	PE
En-De	7,000	1,000	1,000	114,980	16,519	16,371	✓	✓
En-Zh	7,000	1,000	1,000	115,585	16,307	16,765	✓	✓
Ru-En	7,000	1,000	1,000	82,229	11,992	11,760	✓	
Ro-En	7,000	1,000	1,000	120,198	17,268	17,001	✓	
Et-En	7,000	1,000	1,000	98,080	14,423	14,358	✓	
Ne-En	7,000	1,000	1,000	104,934	15,144	14,770	✓	
Si-En	7,000	1,000	1,000	109,515	15,708	15,821	✓	

Table 1: Statistics of the data used for Task 1 (DA) and Task 2 (PE). The number of tokens is computed based on the source sentences.

download.[5] More specifically, the following types of labels were produced:

- **Source side:** Each word in the source side is labelled as OK (correctly translated) or BAD (caused a translation error).
- **Target side:** Each word in the target side is labelled as OK (a correct translation) or BAD (should be replaced or deleted). Additionally, we consider gap 'tokens' at the beginning of the sentence, at the end and between each two words. They are labelled OK if no word should be inserted in that position (according to the post-edited version), and BAD otherwise.

In order to obtain the labels, we first align source and MT using the IBM Model 2 alignments from FastAlign (Dyer et al., 2013), and compute edit distances between the generated and post-edited translations with TERCOM, using default settings and disabled shifts.

2.3 Task 3: Predicting document-level MQM

This task consists in finding document-level translation errors and estimating a quality score according to the amount of minor, major, and critical errors present in the translation. The predictions are compared to a ground-truth obtained from annotations produced by crowd-sourced human translators from Unbabel community.

Each document contains zero or more errors, annotated according to the MQM taxonomy[6], and may span one or more tokens, not necessarily contiguous. Errors have a label specifying their type, such as wrong word order, missing words, agreement, etc. They provide additional information, but do not need to be predicted by the systems. Additionally, there are three severity levels for errors: *minor* (if it is not misleading nor changes meaning), *major* (if it changes meaning), and *critical* (if it changes meaning and carries any kind of implication, possibly offensive).

Figure 1: Example of fine-grained document annotation. Spans in the same color belong to the same annotation. Error severity and type are not shown for brevity.

Figure 1 shows an example of fine-grained error annotations for a sentence. Note that there is an annotation composed by two discontinuous spans: a whitespace and the token *Grip* — in this case, the annotation indicates wrong word order, and *Grip* should have been at the whitespace position.

Document-level scores were then generated from the word-level errors and their severity using the method described in Sanchez-Torron and Koehn (2016, footnote 6). Namely, denoting by n the number of words in the document, and by n_{min}, n_{maj}, and n_{cri} the number of annotated minor, major, and critical errors, the final quality scores were computed as:

$$\text{MQM} = 1 - \frac{n_{minor} + 5n_{major} + 10n_{crit}}{n} \quad (1)$$

[5] https://github.com/deep-spin/qe-corpus-builder

[6] Multidimensional Quality Metrics; see http://www.qt21.eu/mqm-definition/definition-2015-12-30.html for details.

Note that MQM values can be negative if the total severity exceeds the number of words.

As this year's dataset, we reused the training data from previous years, adding the test sets from 2018 and 2019 to the training set, keeping the same development set from 2019, and released a new test set. The documents are short product title and descriptions in English, extracted from the Amazon Product Reviews dataset (McAuley et al., 2015; He and McAuley, 2016) (Sports and Outdoors category). The documents were machine translated into French using a state of the art online neural MT system. The dataset statistics are presented in Table 2.

3 Baseline systems

Sentence-level baseline systems: For Tasks 1 and 2, both word and sentence-level, we used the LSTM-based Predictor-Estimator approach (Kim et al., 2017), implemented in OpenKiwi (Kepler et al., 2019b). The Predictor model was trained on the same parallel data as the NMT systems for each language pair (made available at the task website),[7] while the the Estimator was trained on the $7,000$ QE labelled data for each task.

Word-level baseline systems: For Task 2, we also used the Predictor-Estimator as above, but it was trained to predict jointly word-level tags and sentence-level scores.

Document-level baseline system: For Task 3, similarly as last year, we used a baseline which treats sentences independently and casts the problem as word-level QE, such that all words and gaps within an error span are given the tag BAD. We then trained a Predictor-Estimator model, regrouping any contiguous sequence of tokens tagged as BAD in a single error annotation. In order to get MQM scores, instead of computing the value according to its definition, we compute it simply as 1 minus the the ratio of BAD tags.

4 Participants

Table 3 lists all participating teams submitting systems to any of the tasks, and Table 4 report the number of successful submissions to each of the sub-tasks and language pairs. Each team was allowed up to two submissions for each task variant and language pair. In the descriptions below, participation in specific tasks is denoted by a task identifier (T1 = Task 1, T2 = Task 2, T3 = Task 3).

Bergamot-LATTE (T1): Bergamot-LATTE submitted two systems to the two variants of sentence-level predictions: (i) a black-box approach based on pre-trained representations; (ii) an unsupervised glass-box approach that leverages information extracted from the neural MT system. The black-box model consists of stacking a 2-layer multilayer perceptron on the vector representation of the CLS token from the contextualised representation from XLM-R(Conneau et al., 2020), using both the source and the target sentences as input. The glass-box approach explores the best-performing unsupervised quality indicators presented in Fomicheva et al. (2020c) that rely on uncertainty quantification based on the Monte Carlo dropout method: D-TP and D-Lex-Sim.

Bergamot (T1, T2): Bergamot explores recent work on glass-box QE that exploits NMT output distribution and attention to capture uncertainty as a proxy to MT quality. Specifically, they use three groups of unsupervised quality indicators described in Fomicheva et al. (2020c) as features for a regression model.

Bering Lab (T2): Bering Lab proposes a fine-tuned version of a pre-trained XLM-R model. The model is first trained on a huge artificial QE data that is created by (i) translating a parallel corpus with an OpenNMT system; and (ii) using the TER tool to produce artificial labels for both word- and sentence-levels. The model is then fine-tuned using the shared task's data. For predictions at word-level, the final hidden vector of each token, including the <S>, is fed into a linear layer with sigmoid activation in order to predict the probability of each of these token to be BAD. Quality labels for tokens and gaps are predicted separately with two distinct binary classification layers. For predictions at sentence-level, the final hidden vector of the first <S> token, considered as a pooled representation, is fed into two linear layers with $tanh$ activation. Submitted predictions are results of an ensemble of 5 models trained with different seeds: averaged predictions for sentence-level, and majority voting for word-level.

[7] http://statmt.org/wmt20/quality-estimation-task.html

	Documents			Sentences			Tokens		
	Train	Dev	Test	Train	Dev	Test	Train	Dev	Test
En-Fr	1,448	200	180	8,592	1,301	895	189,735	28,092	18,545

Table 2: Statistics of the data used for Task 3. The number of tokens is computed based on the source sentences.

ID	Participating team	
Bergamot-LATTE	University of Sheffield & Imperial College London, UK & Johns Hopkins University & Facebook AI, US & University of Tartu, Estonia	(Fomicheva et al., 2020b)
Bergamot	University of Tartu, Estonia	(Fomicheva et al., 2020b)
Bering Lab	Bering Lab, Republic of Korea	(Lee, 2020)
Elturco.AI	Elturco AI, Turkey	–
FVCRC	Nagoya University, Japan & University of Sydney, Australia	(Zhou et al., 2020)
HW-TSC	Huawei Translation Services & East China Normal University, China	(Wang et al., 2020a)
IST & Unbabel	Instituto Superior Técnico Lisbon & Unbabel, Portugal	(Moura et al., 2020)
JXNU-CCLQ	Jiangxi Normal University, China	–
Mak	University of Wolverhampton, UK	–
NICT Kyoto	National Institute of ICT, Japan	(Rubino, 2020)
NiuTrans	Northeastern University & NiuTrans Reasearch, China	(Hu et al., 2020)
NJUNLP	Nanjing University, China	(Cui et al., 2020)
Papago	KAIST & Naver, Republic of Korea	(Baek et al., 2020)
RTM	Boğaziçi University, Turkey	(Biçici, 2020)
TMUOU	Osaka University & Tokyo Metropolitan University, Japan	(Nakamachi et al., 2020)
Tencent Inc.	Tencent Inc, China	(Wang et al., 2020b)
TransQuest	University of Wolverhampton, UK	(Ranasinghe et al., 2020)
WL Research	WL Research, US, Canada and Turkey	(Kane et al., 2020)
XC	Imperial College London, UK	–

Table 3: Participants in the WMT20 Quality Estimation shared task.

Task/LP	# submission
Task 1 – Sent-level Direct Assessment	**747**
Multilingual	43
English-German	132
English-Chinese	146
Romanian-English	150
Nepali-English	56
Estonian-English	68
Sinhala-English	74
Russian-English	78
Task 2 – Post-Editing Effort	**435**
English-German (sent-level)	131
English-Chinese (sent-level)	235
English-German (word-level)	38
English-Chinese (word-level)	31
Task 3 – Document-Level QE	**192**
English-French (annot.)	97
English-French (score)	95
Total	**1374**

Table 4: Number of submissions to each sub-task and language-pair at the WMT20 Quality Estimation shared task. In the results (Section 5) we only report the top two submissions per team for each task and language pair.

Elturco.AI (T2): Elturco.AI uses a generative model and a discriminative model, inspired by Electra (Clark et al., 2020). The two models are jointly trained on a parallel corpus, in order to create increasingly difficult artificial samples for quality estimation. The generative model consists of a transformer encoder and two transformer decoders, for forward and backward direction. In addition to predicting tokens, it is also trained to predict gap locations on the target side given the source sentence and left and right contexts on the target side. Distorted translations are generated by sampling on generator outputs on token and gap locations, which can be shorter or longer than the original translation. The distorted translations are compared to original translations for generating token and gap tags. The discriminator, a transformer encoder-decoder with full attention mask on the decoder side, is trained to predict the generated tags given the source and distorted translation. Once trained, the discriminator is fine-tuned on the actual quality estimation dataset.

FVCRC (T1): FVCRC's system builds on BERTScore, a text generation evaluation system based on pretrained BERT contextual embeddings, originally for Metrics tasks. By

using pre-trained multilingual BERT-based model, they experiment with BERTScore on QE tasks. Without reference translations, it makes more errors in terms of word (or sub-word) alignments when perform greedy matching on pairwise cosine similarity, which is believed to be main cause of its drop of performance in QE tasks. They introduce GIZA++ word (subword) alignments and n-grams similarity matching to tackle misalignments and sentence perplexity of candidate translation as additional information to the evaluation score. Otherwise, the default setting of BERTScore (Zhang* et al., 2020) is used: pre-trained bert-base-multilingual-cased and xlm-mlm-100-1280 for embedding extraction, with a single model. This system is not trained on human labels (DA) and is not optimised on additional data.

HW-TSC (T2): HW-TSC submissions follows the Predictor-Estimator architecture (Kim et al., 2017), with a pre-trained Transformer as Predictor, and task-specific classifiers and regressors as Estimators. HW-TSC uses a unified model to solve both word- and sentence-level tasks, trained under multi-task learning. To improve the transfer-learning efficiency across tasks while preventing over-fitting, a Bottleneck Adapter Layer (Houlsby et al., 2019) is added to the Transformer after the self-attention and the feedfoward layers, while keeping the original parameters of the Transformer model fixed.

IST & Unbabel (T1, T2, T3): IST & Unbabel submitted two systems per task variant: OPENKIWI-BASE and KIWI-GLASS-BOX-ENSEMBLE for predicitons at both word- and sentence-levels; KIWI-DOC and KIWI-DOC-IOB for document-level predictions. OPENKIWI-BASE is based on the re-implementation of the Predictor-Estimator architecture (Kim et al., 2017) available in OpenKiwi (Kepler et al., 2019b): the Predictor model is replaced with pre-trained contextualised representations (such as BERT or XLM-R) and the bi-LSTM Estimator is replaced by linear layers. KIWI-GLASS-BOX-ENSEMBLE is similar to OPENKIWI-BASE with a bottleneck layer introduced in the Estimator in order to concatenate the features extracted from the Predictor, with sentence-level uncertainty features extracted from the NMT system provided by the shared task. Those glass-box features are based on work by Fomicheva et al. (2020c) and exploit entropy measures at prediction time. Unlike OPENKIWI-BASE, the KIWI-GLASS-BOX-ENSEMBLE model is trained for source, target and sentence level predictions simultaneously, using a multi-task learning approach. KIWI-DOC is the same as in Kepler et al. (2019a) while KIWI-DOC-IOB frames the task of annotating as Name Entity Recognition task: the severity annotations are projected to tags in IOB format (‘O‘, ‘B-major‘, ‘I-major‘, ‘B-critical‘, etc.) and the model is trained with a CRF output layer to enforce correctness of the tag-sequence at prediction time. The predicted tags are converted into annotations without the resort to a grouping and labelling heuristic.

JXNU-CCLQ (T1): JXNU-CCLQ proposes a model composed of a Transformer bottleneck layer and a bidirectional LSTM. The parameters of the Transformer bottleneck layer are first optimised with a bilingual parallel corpus, and the entire model is then fine-tuned on the training quality labelled dataset of the shared task. At test time, the translation outputs, which are estimated with teacher forcing and special masking, are put together with the source sentences and put through a unified neural network model to predict the quality of the translations.

Mak (T1): Mak represents the source and its translation sentence pairs as a set of 70 black-box sentence-level features extracted with Quest++(Specia et al., 2015), using the resources used to train the English-Russian NMT system (Ng et al., 2019). Those features are then fitted into a support vector regressor with default settings.

NICT Kyoto (T2): The English–German and English-Chinese sentence-level QE systems for Task 2 are ensembles of pre-trained cross-lingual language models (XLM) (Conneau and Lample, 2019), fine-tuned in a multi-task fashion with two linear output layers for sentence and word-level quality estimation. A total of 8 XLM models with various masking hyper-parameters were domain-adapted

using a subset of the additional resources provided by the QE shared task organisers, as well as a subset of the WikiMatrix corpus [2]. The translation language model training approach (TLM) was used before fine-tuning the XLM models for the QE task, complemented with a novel self-supervised learning task which aims to model errors inherent to machine translation outputs.

NiuTrans (T1, T2, T3): For Task 1, NiuTrans explored the combination of pre-trained models and multi-task learning. They used three different model settings, including multilingual-bert (∼200M parameters), xlm-roberta-base (∼300M parameters) and xlm-roberta-large (∼600M parameters). They continued pre-training all models on the WMT dataset and utilised task adaptive pre-training to further boost the models' performance. The output of different models was combined using a weighting scheme to get final predictions. For Task 2, an ensemble of 10 transformer-based predictor-estimator models was used, with multi-task training for the word-level tasks. Each single model contains 10M parameters. They also submitted an ensemble result of multilingual-bert and xlm-roberta-base for sentence-level scoring tasks. For Task 3, they used an ensemble of 8 predictor-estimator models and multi-task training for the word-level subtask. The single model contains 150M parameters. For the scoring subtask, they explored an ensemble of linear regression models and pre-trained models. They also used WMT 2014 English-French dataset for fine-tuning.

NJUNLP (T2): This system is an ensemble using NuQE and QUETCH models (Kepler et al., 2019b), as well as the QE Brain model (Fan et al., 2019). In addition to these pre-existing models, the ensemble also uses a masked version of the QE Brain, where some tokens in the translation are masked during training, and a masked language model (Devlin et al., 2018). For sentence-level, the different models are used as feature extractors, which are used as inputs of a dense layer to produce the predictions. For word-level, they use majority voting to ensemble the different models.

Papago (T1, T3): Papago's submission for Task 1 En-De is an ensemble of three models based on pre-trained contextualised representations: multilingual BERT (mBERT), XLM-Masked-Language-Modelling (XLM-MLM), and XLM-Causal-Language-Modelling (XLM-CLM). Three scores were produced from these models: an extension of BERTScore using the multilingual BERT model, SentenceBERT score (Reimers and Gurevych, 2019), and target (German) language model score using a pre-trained GPT-2 model. Additionally, the scores were computed for synthetic data created using WMT News translation data by randomly performing different methods, including swapping word order, omiting words or repeating phrases. The three models are pre-trained from these data in a multi-task regression setting. Lastly, these pre-trained models are fine-tuned using the QE corpus. For Task 3, the submitted system uses an ensemble of four models leveraging either multilingual BERT or XLM. The training scheme is very task-oriented: erroneous sentence pairs and their pseudo-MQM scores are generated from Europarl and this QE task's training corpus.

RTM (T1, T2): For Task 1 and Task 2's sentence-level prediction, the RTM model treats QE as a parallel semantic similarity prediction task within machine translation performance prediction (MTPP) or a monolingual semantic similarity when the source or the target language are unknown or have scarce resources. En-De and Ru-En were modelled as parallel MTPP and the rest as monolingual MTPP using only the English side of the training and development datasets. Machine learning algorithms including ridge regression, SVR, and regression trees were used and the submissions were constrained to the resources provided. RTM selects a subset of parallel and monolingual text for each translation direction.

TMUOU (T1): TMUOU proposes an ensemble of four regression models based on XLM-R large: model 1 uses the final hidden vector of the CLS token; model 2 concatenates the feature of model 1 with the mean of the final hidden vector of each token; models 3 and 4 are based on models 1 and 2, respectively, but

adds a special token for language identification at the beginning of each sentence. The ensemble model is a gradient boosting regressor that features the predictions of these four models, the sentence probability of the target translation system, and one-hot vectors that indicate both the source and target languages.

Tencent (T2): Tencent-TTL's submission for the sentence-level Task 2 use a predictor-estimator model. They use two predictors as feature extractors: a transformer trained with WMT provided parallel corpus and a fine-tuned cross-lingual language model (XLM). For the XLM-based predictor, it produces two kinds of contextual token representations, i.e., masked representations and non-masked representations. For transformer-based predictor, only the non-masked representation is produced. The estimator was trained with LSTM or Transformer. Finally, they ensembled the systems with different models and the same model with different parameters using logistic regression to produce a single sentence-level prediction.

TransQuest (T1, T2): TransQuest proposes two architectures: MONOTRANSQUEST and SIAMESETRANSQUEST, both using pre-trained XLM-R large transformer model. The MONOTRANSQUEST architecture uses the computed CLS token pooled representation from a single transformer model and uses it as input of a softmax layer that predicts the quality score of the translation. The SIAMESETRANSQUEST architecture encodes both the source sentence and its translation with two separate XLM-R transformer models. For each transformer model, the it computes the mean of all output vectors of the input words, and applies the cosine similarity measure between the two outputs. The final submission is an ensemble of the two architectures.

WL Research (T1): WL's NUBIA method has three modules: a neural feature extractor, an aggregator and a calibrator. The feature extractor consists of different transformer-based architectures fine-tuned on relevant tasks of language evaluation such as semantic similarity (RoBERTa model fine-tuned on STS-B), logical inference (RoBERTa fine-tuned on MNLI) and sentence likelihood (GPT2 perplexity score). The aggregator uses the features extracted by the transformers as well as non-neural features such as hypothesis sentence length and is trained to predict the quality of the hypothesis sentence. These features are then used to train a 10 hidden layer neural network. Given that NUBIA takes as input sentences in English, as pre-processing step, Google Translate was used to translate either the non-English candidate or source to have both in English.

XC (T1): This was a multilingual system trained using TransQuest (with BERT embeddings bert-base-multilingual-cased) and data for all language pairs concatenated. An attempt was also made to use project the BERT source and target sentence embeddings into a space where they are highly correlated using CCA (Canonical Correlation Analysis) followed by an MLP regressor trained to predict the quality score, but this did not perform as well as a vanilla TransQuest.

5 Results

5.1 Task 1

Submissions for Task 1 are **evaluated** against the true z-normalised direct assessment label using Pearson's r correlation score as primary metric. This is what was used for ranking system submissions. Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE) were also computed as secondary metrics. Statistical significance on Pearson r was computed using William's test.[8]

Table 5 summarises the results for all language pairs, as well as the multilingual variant, in terms of Pearson's r correlation with direct assessments, ranking systems by their average performance for all language pairs (using 0 as Pearson score for other languages). In the Appendix, Tables 11, 12, 13, 14, 15, 16, 17 and 18 provide the detailed results for all language pairs and the multilingual variant, ranking participants by their performance for each of these cases. The detailed tables show a striking difference in performance by Pearson scores versus MAE/RMSE, especially for the top systems. This requires further investigation.

Best performers The two top performing systems, TransQuest and Bergamot-LATTE (black-

[8] https://github.com/ygraham/mt-qe-eval

Model	Si-En	Ne-En	Et-En	Ro-En	En-De	En-Zh	Ru-En	Multi
TransQuest	**0.68**	**0.82**	**0.82**	**0.91**	**0.55**	**0.54**	**0.81**	**0.72**
Bergamot-LATTE (black-box)	**0.68**	0.81	**0.83**	**0.91**	**0.54**	0.53	0.80	**0.72**
IST and Unbabel (Kiwi-glass-box-ensemble)	0.64	0.79	0.77	0.89	0.52	0.49	0.77	0.67
TMUOU	0.67	0.78	0.79	0.90	0.48	0.44	0.78	0.69
XC	0.63	0.78	0.76	0.88	0.47	0.47	0.78	-
WL Research	0.58	0.69	0.64	0.82	0.25	0.30	0.60	0.55
Bergamot	0.56	0.66	0.68	0.80	0.48	0.43	-	-
Bergamot-LATTE (glass-box)	0.51	0.60	0.64	0.69	0.26	0.32	-	0.49
IST and Unbabel (OpenKiwi-base)	0.56	0.60	0.69	0.71	0.27	0.35	-	0.58
BASELINE	0.37	0.39	0.48	0.68	0.15	0.19	0.55	0.38
FVCRC	0.39	0.49	-	0.65	0.11	0.08	0.40	-
RTM	0.54	-	0.61	0.70	-	0.26	-	-
Shrangin ‡	-	-	-	0.85	-	-	-	-
Mak	-	-	-	-	-	-	0.54	-
Papago	-	-	-	-	0.50	-	-	-
aj54 ‡	-	-	-	-	-	0.44	-	-
JXNU-CCLQ	-	-	-	-	-	0.43	-	-
jackielo ‡	-	-	-	-	-	-	0.41	0.46
zhanghuimeng ‡	-	-	-	-	0.39	-	-	-
DexinWang ‡	-	-	-	-	0.25	-	-	-
Hancheng-Deng ‡	-	-	-	-	0.17	-	-	-
NiuTrans †	0.70	0.83	0.83	0.92	0.56	0.55	0.82	0.73

Table 5: Pearson correlation with direct assessments for the submissions to WMT20 Quality Estimation Task 1. For each language pair, results marked in bold correspond to the winning submissions, as they are not significantly outperformed by any other system according to the Williams Significance Test (Williams, 1959). Baseline systems are highlighted in grey; † indicates teams that have been identified as having submitted more systems than the allowed limit to the leaderboard; ‡ indicates Codalab username of participants from whom we have not received further information.

box) perform the same or very closely for all language pairs. Both make use of the XML-R large pre-trained representations, and ensembles. This is clearly a booster, as these systems achieve almost double the correlation of the baseline. Note that the baseline also uses pre-trained word embeddings, but these are obtained using much smaller datasets: those used to train the NMT models for each respective language pair.

Except for a few systems for some language pairs, the vast majority of submissions outperform the baseline system, often by a large margin, except for Russian-English which had fewer submissions and where 1/3 of the systems are below the baseline. It is hard to make any conclusions about this difference across languages as Russian-English systems that are below the baseline did not submit systems for other languages. In relative terms, the improvement over the baseline for top systems in this language is similar to the other language pairs. The range of performances is remarkably wide, with the winning systems often doubling the Pearson correlation of the bottom pack, notably for English-Chinese and English-German.

To gain a better understanding in the performance of different QE approaches for different language pairs, Figure 2 shows the scatter plots for the baseline and the best performing system for each language pair. Note the remarkable difference in correlation between the baseline and the top performers across languages. In the figures, we can visualise the substantial gains are achieved, largely due to the use of strong pre-trained representations.

High-resource performance MT quality for the high-resource language pairs, in particular English–German, was the most challenging to predict. As discussed in Fomicheva et al. (2020a), the MT outputs for this language pair have little variability in terms of perceived MT quality. The vast majority of translations were assigned high scores during DA evaluation, which makes it difficult to capture any meaningful variation between the DA scores. We observe that the results for HTER prediction for this language pair are more positive, a difference which we discuss in Section 6.

Low-resource performance Interestingly, the results for the low-resource language pairs, Sinhala–English and Nepali–English, are comparable with the rest. The fact that the performance of the winning approaches based on multilingual pre-trained representations does not degrade for the low-resource language pairs is worth noticing. It could indicate that: (*i*) the source language does

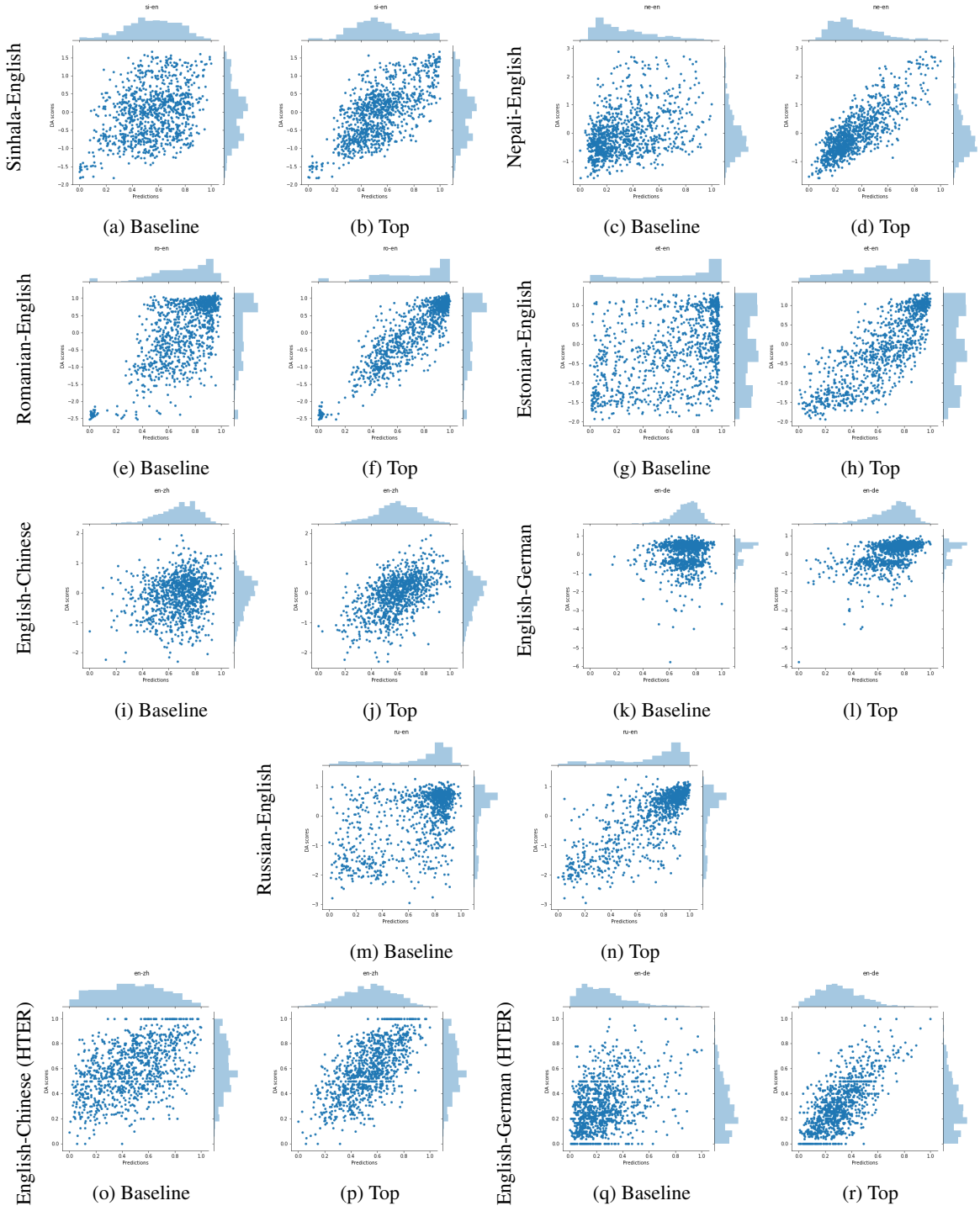


Figure 2: Scatter plots for the predictions against true scores for the baseline and top-performing systems. Sub-figures (a) through (n) show systems trained on direct assessment, while sub-figures (o) through (r) show systems trained on HTER. Predictions are scaled to $[0..1]$.

not have as high an impact on QE as the target language, which has been previously observed as a problem for QE with *partial-input* experiments (Sun et al., 2020); (*ii*) shared supervision on the target side is beneficial, and thus having more training data into English inherently benefits multiple languages; or (*iii*) the distribution of scores is more balanced for low- and medium-resource languages, which makes the task easier. To shed more light on this, for future shared tasks, we recommend having more low resource languages as the target language for QE, or more data with shared languages on the source.

High correlations Finally, for some language pairs the performance of the top system is very high, particularly for Romanian–English (Pearson $r = 0.91$). As shown in Figure 2f, there is a number of very low-quality sentences that the QE systems are able to successfully detect. By inspecting those cases, we find that they correspond to 'hallucinated' outputs from the Romanian–English MT system that do not have anything to do with the original sentences. Detecting such cases should be trivial for QE systems, which explains the particularly high correlation values for this language pair. This highlights a possible issue with using Pearson correlation to evaluate the performance of QE systems: strong correlations can be achieved by having an over-representation extreme values (i.e. really bad or really good translations), and bad correlations can be an artefact of the lack of representation of extreme values (as in the case of English-German).

5.2 Task 2

Sentence-level post-editing effort: For this task variant, **evaluation** was performed against the true HTER label using the same metrics as in Task 1, with Pearson's r correlation score as the primary metric. Statistical significance on Pearson r was computed using the William's test.

Table 6 summarises the results for English–German and English–Chinese tracks, ranking systems by their average performance for the two language pairs (with 0 as Pearson score for languages without systems). In the Appendix, Tables 19 and 20 show the detailed evaluation results for the two language pairs, ranking participating systems best to worst using Pearson's r correlation as primary key.

For English–German, the two top performing systems, HW-TSC and Bering Lab, are substantially ahead of the other participants' systems, with a considerable advantage for HW-TSC, which is the top system with statistical significance. For English–Chinese, Tencent and IST/Unbabel glass-box system were the top performing systems and neither outperforms the other; for this language pair, the range of Pearson scores achieved by participants' systems is much narrower than for English-German. Finally, for both language pairs, we see that all submissions outperform the baseline system by a large margin, most prominently for English–German.

Word-level errors For this task, the primary **evaluation** metric is Matthews correlation coefficient (MCC, Matthews, 1975). We also report the F_1-scores for the OK and BAD classes. Similarly to the 2019 edition, we evaluate separately the source and target side, with the latter including predictions on actual target words as well as gaps. The word-level results for Task 2 are summarised in Tables 7 and 8, ordered by the MCC metric on target errors.

The number of submissions per language pair was different, which limits any conclusions that can be made with respect to general rankings of systems. For English–German, the findings are similar to the sentence-level task: the Bering Lab and HW-TSC teams are the top performing systems by a great margin, with the former better at predicting source side errors and the latter slightly better at predicting target side errors. For English–Chinese, the range of scores is narrower, with HW-TSC, NICT Kyoto, and IST/Unbabel all performing very closely (with HW-TSC on top). For both language pairs, all systems performed above the baseline, and we also see that the scores for the source side are substantially lower than the target side.

5.3 Task 3

MQM score estimation For the document-level estimation task, submissions are evaluated in terms of Pearson's correlation r, as in Tasks 1 and 2, between the true and predicted document-level scores. Participants results are shown in Table 9. This task attracted fewer participants than the other two, probably because it is more complex. Papago has the best results, with a considerable gap to the IST/Unbabel, which in turn also were well ahead of the baseline.

Fine-grained annotations Fine-grained annotations are evaluated as follows. For each error anno-

Model	En-De	En-Zh
IST and Unbabel (Kiwi-glass-box)	0.633	**0.651**
NJUNLP	0.618	0.642
NICT Kyoto	0.615	0.643
Bergamot	0.613	0.613
IST and Unbabel (OpenKiwi-base)	0.531	0.593
TransQuest	0.499	0.612
BASELINE	0.392	0.506
HW-TSC	**0.758**	-
Bering Lab	0.723	-
Tencent Inc.	-	**0.664**
niuniuniu ‡	-	0.569
aj54 ‡	-	0.552
zhanghuimeng ‡	0.494	-
DexinWang ‡	0.402	-
NiuTrans †	0.649	0.675

Table 6: Pearson correlation with direct assessments for the submissions to WMT20 Quality Estimation Task 2. For each language pair, results marked in bold correspond to the winning submissions, as they are not significantly outperformed by any other system according to the Williams Significance Test (Williams, 1959). Baseline systems are highlighted in grey; † indicates teams that have been identified as having submitted more systems than the allowed limit to the leaderboard; ‡ indicates Codalab username of participants from whom we have not received further information.

	Target Side			**Source Side**		
Model	MCC	F_1-BAD	F_1-OK	MCC	F_1-BAD	F_1-OK
Bering Lab	0.597	0.662	0.935	0.454	0.609	0.818
HW-TSC	0.583	0.644	0.938	0.523	0.649	0.875
NICT Kyoto	0.485	0.568	0.916	0.353	0.537	0.806
IST and Unbabel (Kiwi-glass-box)	0.465	0.550	0.916	0.349	0.535	0.801
NJUNLP	0.451	0.498	0.929	–	–	–
IST and Unbabel (OpenKiwi-base)	0.432	0.522	0.909	0.324	0.516	0.799
Elturco.AI	0.423	0.520	0.887	–	–	–
BASELINE	0.358	0.468	0.879	0.266	0.477	0.779
NuiTrans †	0.500	0.581	0.916	0.347	0.532	0.806

Table 7: Official results of the WMT20 Quality Estimation Task 2 word-level for the **English–German** dataset. Baseline systems are highlighted in grey; † indicates teams have been identified as having submitted more systems than the allowed limit to the leaderboard.

	Target Side			**Source Side**		
Model	MCC	F_1-BAD	F_1-OK	MCC	F_1-BAD	F_1-OK
HW-TSC	0.587	0.714	0.866	–	–	–
NICT Kyoto	0.582	0.704	0.878	0.336	0.668	0.669
IST and Unbabel (OpenKiwi-base)	0.575	0.706	0.850	0.287	0.705	0.410
IST and Unbabel (Kiwi-glass-box)	0.567	0.701	0.842	0.287	0.705	0.403
NJUNLP	0.551	0.672	0.877	–	–	–
BASELINE	0.509	0.658	0.849	0.270	0.682	0.547
NuiTrans †	0.610	0.723	0.887	0.308	0.666	0.639

Table 8: Official results of the WMT20 Quality Estimation Task 2 word-level for the **English–Chinese** dataset. Baseline systems are highlighted in grey; ‡ indicates teams have been identified as having submitted more systems than the allowed limit to the leaderboard.

tation a_i^s in the system output, we look for the gold annotation a_j^g with the highest overlap in number of characters. The precision of a_i^s is defined by the ratio of the overlap size to the annotation length; or 0 if there was no overlapping gold annotation. Conversely, we compute the recall of each gold annotation a_j^g considering the best matching annotation a_k^s in the system output,[9] or 0 if there was no overlapping annotation. The document precision and recall are computed as the average of all annotation precision in the corresponding system output and recalls in the gold output; and therewith we compute the document F_1. The final score is the unweighted average of the F_1 for all documents.

The annotation scores are shown in Table 10. Only one participant, IST/Unbabel submitted valid results, but still better than the baseline.

6 Discussion

In what follows, we discuss the main findings of this year's shared task based on the goals we had previously identified for it.

General progress. Overall, participating systems achieved very promising results, with the best performing submissions showing moderate to strong correlation for sentence-level DA and HTER prediction tasks. One reason for high correlation levels is likely to be that top performing systems are based on pre-trained representations. Like in other NLP tasks, for QE it had already been shown to substantially improve the results over models that do not use such representations, with heavier pre-trained embeddings contributing substantially more (Kepler et al., 2019a). Strong pre-trained embeddings such as XLM-R were used by most submissions this year.

When interpreting the results for all tasks, it should be noted that most of the participants use extremely resource-heavy systems, ensembles of multiple models with more than 500M parameters, which could make them difficult to use in practice. Reporting the number of parameters could be a good practice for the future.

Comparison to previous years submissions are not possible as they use very different datasets, except for Task 3, where a new test set was collected from the same initial larger dataset, but the training data is virtually the same. For the fine-grained version of the task, results are on par with last year (0.48 F1), while for the scoring variant the results this year are more encouraging: while the baseline remains similar (Pearson = 0.39 this year and 0.35 last year), the top system is significantly better this year: 0.57 Pearson instead of 0.37 last year.

Unfortunately, the document-level task still attracts very few participants, being naturally more difficult to model. However, document-level translation quality is a growing concern in the MT community, and we believe it is interesting that this task continues to exist, possibly with a different dataset and format, in the next editions.

Comparison between HTER and DA. Compared to the results from the previous editions of this shared task, participating systems show overall higher correlation with DA labels. Besides the QE systems getting much stronger, DA labels might be easier to predict, as HTER is a semi-automatic metric and may suffer from the same issues as TER, as it does not capture to what extent the overall quality of the sentence is affected by MT errors. We should note, however, that for the language pairs selected for post-editing this year (English–German and English–Chinese) the correlation is higher for HTER. A possible reason is a very skewed output distribution of the DA scores for these particular language pairs.

HTER and DA annotation capture different aspects of translation quality. In fact, as shown in Fomicheva et al. (2020a), the correlation between the two types of scores is fairly low. An interesting question is whether the approaches that perform best for predicting DA also achieve the best results for HTER. Figure 3 plots sentence-level Pearson correlation with HTER and direct assessments for the systems that participated in both tasks. While the systems with the highest and the lowest ranks are the same, results change considerably for the systems in the middle. Specifically, TransQuest is one of the winning submissions for the prediction of DA, but is outperformed by the submissions that use glass-box features, i.e. Bergamot and IST and Unbabel (Kiwi-glass-box) for the HTER task.

Multilingual approaches. Most of the participating approaches rely on pre-trained multilingual representations and use the provided data annotated with quality labels for fine-tuning. This shows the potential for multilingual prediction in these systems making them much more appealing in prac-

[9] Notice that if a gold annotation a_j^g has the highest overlap with a system annotation a_i^s, it does not necessarily mean that a_i^s has the highest overlap with a_j^g.

Model	**Pearson** r	MAE	RMSE
Papago	0.573	15.611	23.327
IST and Unbabel (Kiwi-doc-iob)	0.475	17.127	25.530
BASELINE	0.389	19.939	26.608
NiuTrans †	0.494	20.607	24.258

Table 9: Official results of the WMT20 Quality Estimation Task 3 scoring for the **English–French** dataset. Baseline systems are highlighted in grey; † indicates teams have been identified as having submitted more systems than the allowed limit to the leaderboard.

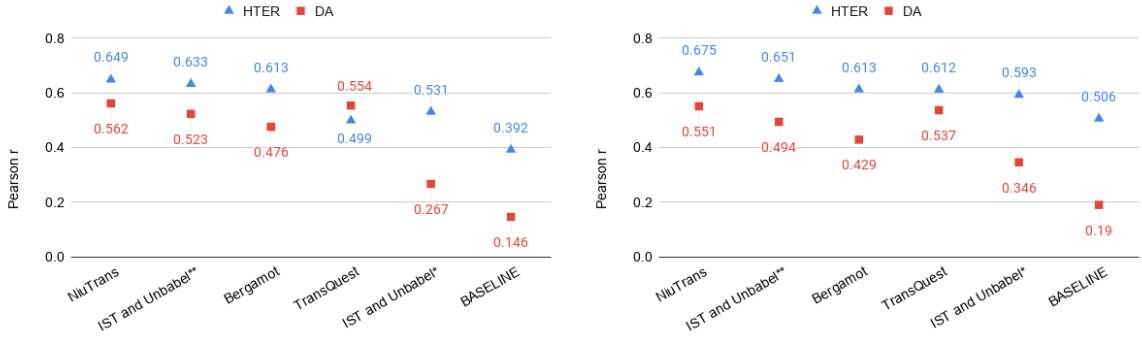


Figure 3: Pearson correlation for the systems that participated in both Task 1 and Task 2 at sentence level for English-German (left) and English-Chinese (right). OpenKiwi-base and Kiwi-glass-box submissions are marked with $*$ and $**$ respectively.

Model	F1
IST and Unbabel (Kiwi-doc)	0.472
BASELINE	0.416
NiuTrans †	0.418

Table 10: Official results of the WMT20 Quality Estimation Task 3 annotation for the **English–French** dataset. Baseline systems are highlighted in grey; † indicates teams have been identified as having submitted more systems than the allowed limit to the leaderboard.

tice where having dedicated systems for each language pair may be infeasible. However, in the task most submissions built models specific to each language pair, and then submitted their predictions to the multilingual task. A notable exception is the Bergamot-LATTE team, where a single prediction model was trained for all languages.

Influence of source-language document-level context. To investigate the utility of document-level information, we offered to participants the title of the Wikipedia article where the sentences were extracted for Tasks 1 and 2. However, no participating system requested these additional labels, and therefore this remains an open question.

Applicability of NMT model information. Multiple submissions use glass-box features based on the information extracted from the NMT system in an unsupervised manner (Bergamot-LATTE), in a regression setting (Bergamot) or in combination with pre-trained representations (IST and Unbabel). Results show the potential of this approach. Although substantially outperformed by the top submissions that use pre-trained representations trained with very large amounts of data, glass-box approaches beat the baseline, which use the same amount of training data as the NMT system, by a large margin. These approaches might offer a better trade-off between accuracy and efficiency for cases where the NMT model is accessible.

New publicly available benchmarks. Creating the multi-language, multi-label dataset for this year's edition was a significant joint effort from various institutions, and we hope it will be useful for researchers in QE as well as in related areas. For example, Task 2 data was also used for the WMT20 Automatic Post-Editing task. We hope to continue adding data to this collection following the same principles, and that others will also contribute by adding other languages to it in the future. We made all submissions to the task available for

those interested in further analysing the results, investigating approaches for prediction ensembling, among others.

7 Conclusions

This year's edition of the QE Shared Task introduced a number of new elements: the largest number of languages ever, new types of annotation (direct assessment, in addition to labels derived from post-editing and manual error tagging), and number of samples annotated overall. It also attracted the largest number of teams and submissions. We believe the current set of tasks covers a broad enough range of challenges that are far from solved, such as improving performance for languages with skewed distributions, addressing low resource languages, predicting source words that lead to errors, multilingual or language-independent models, etc. In future editions, we hope to keep pushing for progress in these areas.

Acknowledgments

Marina Fomicheva, Frédéric Blain and Lucia Specia were supported by funding from the Bergamot project (EU H2020 Grant No. 825303). André Martins and Erick Fonseca were funded by the P2020 programs Unbabel4EU (contract 042671) and MAIA contract 045909), by the European Research Council (ERC StG DeepSPIN 758969), and by the Fundação para a Ciência e Tecnologia through contract UIDB/50008/2020. We would like to thank Camila Pohlmann and the Unbabel community team for monitoring the post-editing process. We thank IQT Labs for providing the Russian-English dataset for Task 1.

References

Yujin Baek, Zae Myung Kim, Jihyung Moon, Hyunjoong Kim, and Eunjeong Park. 2020. Patquest: Papago translation quality estimation. In *Proceedings of the Fifth Conference on Machine Translation, Volume 2: Shared Tasks Papers*, Online. Association for Computational Linguistics.

Ergun Biçici. 2020. Rtm ensemble learning results at quality estimation task. In *Proceedings of the Fifth Conference on Machine Translation, Volume 2: Shared Tasks Papers*, Online. Association for Computational Linguistics.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems 32*, pages 7059–7069.

Qu Cui, Xiang Geng, Shujian Huang, and Jiajun Chen. 2020. Nju's submission for wmt2020 qe shared task. In *Proceedings of the Fifth Conference on Machine Translation, Volume 2: Shared Tasks Papers*, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Chris Dyer, Victor Chahuneau, and Noah A Smith. 2013. A simple, fast, and effective reparameterization of ibm model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648.

Kai Fan, Jiayi Wang, Bo Li, Fengming Zhou, Boxing Chen, and Luo Si. 2019. "bilingual expert" can find translation errors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6367–6374.

Marina Fomicheva, Shuo Sun, Erick Fonseca, Frédéric Blain, Vishrav Chaudhary, Francisco Guzmán, Nina Lopatina, Lucia Specia, and André F. T. Martins. 2020a. MLQE-PE: A multilingual quality estimation and post-editing dataset. *arXiv preprint arXiv:2010.04480*.

Marina Fomicheva, Shuo Sun, Lisa Yankovskaya, Frédéric Blain, Vishrav Chaudhary, Mark Fishel, Francisco Guzmán, and Lucia Specia. 2020b. Bergamot-latte submissions for the wmt20 quality estimation shared task. In *Proceedings of the Fifth Conference on Machine Translation, Volume 2: Shared Tasks Papers*, Online. Association for Computational Linguistics.

Marina Fomicheva, Shuo Sun, Lisa Yankovskaya, Frédéric Blain, Francisco Guzmán, Mark Fishel, Nikolaos Aletras, Vishrav Chaudhary, and Lucia Specia. 2020c. Unsupervised quality estimation for neural machine translation. *Transactions of the Association for Computation Linguistics*.

Yvette Graham, Timothy Baldwin, Alistair Moffat, and Justin Zobel. 2016. Can machine translation systems be evaluated by the crowd alone. *Natural Language Engineering*, FirstView:1–28.

Francisco Guzmán, Peng-Jen Chen, Myle Ott, Juan Pino, Guillaume Lample, Philipp Koehn, Vishrav Chaudhary, and Marc'Aurelio Ranzato. 2019. The FLORES evaluation datasets for low-resource machine translation: Nepali–English and Sinhala–English. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6098–6111, Hong Kong, China. Association for Computational Linguistics.

Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *proceedings of the 25th international conference on world wide web*, pages 507–517. International World Wide Web Conferences Steering Committee.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. *arXiv preprint arXiv:1902.00751*.

Chi Hu, Hui Liu, Kai Feng, Chen Xu, Nuo Xu, Zefan Zhou, Shiqin Yan, Yingfeng Luo, Chenglong Wang, Xia Meng, Tong Xiao, and Jingbo Zhu. 2020. The niutrans system for the wmt20 quality estimation shared task. In *Proceedings of the Fifth Conference on Machine Translation, Volume 2: Shared Tasks Papers*, Online. Association for Computational Linguistics.

Hassan Kane, Muhammed Yusuf Kocyigit, Ali Abdalla, Pelkins Ajanoh, and Mohamed Coulibali. 2020. NUBIA: Neural based interchangeability assessor for text generation.

Fábio Kepler, Jonay Trénous, Marcos Treviso, Miguel Vera, António Góis, M Amin Farajian, António V Lopes, and André FT Martins. 2019a. Unbabel's participation in the wmt19 translation quality estimation shared task. *arXiv preprint arXiv:1907.10352*.

Fábio Kepler, Jonay Trénous, Marcos Treviso, Miguel Vera, and André F. T. Martins. 2019b. OpenKiwi: An open source framework for quality estimation. In *Proceedings of ACL 2019 System Demonstrations*.

Hyun Kim, Jong-Hyeok Lee, and Seung-Hoon Na. 2017. Predictor-estimator using multilevel task learning with stack propagation for neural quality estimation. In *Proceedings of the Second Conference on Machine Translation*, pages 562–568.

Dongjun Lee. 2020. Two-phase cross-lingual language model fine-tuning for machine translation quality estimation. In *Proceedings of the Fifth Conference on Machine Translation, Volume 2: Shared Tasks Papers*, Online. Association for Computational Linguistics.

Brian W Matthews. 1975. Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta (BBA)-Protein Structure*, 405(2):442–451.

Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 43–52. ACM.

João Moura, Miguel Vera, Daan van Stigt, Fabio Kepler, and André F. T. Martins. 2020. Ist-unbabel participation in the wmt20 quality estimation shared task. In *Proceedings of the Fifth Conference on Machine Translation, Volume 2: Shared Tasks Papers*, Online. Association for Computational Linguistics.

Akifumi Nakamachi, Hiroki Shimanaka, Tomoyuki Kajiwara, and Mamoru Komachi. 2020. Tmuou submission for wmt20 quality estimation shared task. In *Proceedings of the Fifth Conference on Machine Translation, Volume 2: Shared Tasks Papers*, Online. Association for Computational Linguistics.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook fair wmt19 news translation task submission. In *Proc. of WMT*, pages 1–4.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling.

Tharindu Ranasinghe, Constantin Orasan, and Ruslan Mitkov. 2020. Transquest at wmt2020: Sentence-level direct assessment. In *Proceedings of the Fifth Conference on Machine Translation, Volume 2: Shared Tasks Papers*, Online. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Raphael Rubino. 2020. Nict kyoto submission for the wmt'20 quality estimation task: Intermediate training for domain and task adaptation. In *Proceedings of the Fifth Conference on Machine Translation, Volume 2: Shared Tasks Papers*, Online. Association for Computational Linguistics.

Marina Sanchez-Torron and Philipp Koehn. 2016. Machine translation quality and post-editor productivity. *AMTA 2016, Vol.*, page 16.

Lucia Specia, Gustavo Paetzold, and Carolina Scarton. 2015. Multi-level translation quality prediction with quest++. In *ACL-IJCNLP 2015 System Demonstrations*, pages 115–120, Beijing, China.

Shuo Sun, Francisco Guzmán, and Lucia Specia. 2020. Are we estimating or guesstimating translation quality? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6262–6267, Online. Association for Computational Linguistics.

Minghan Wang, Hao Yang, Hengchao Shang, Daimeng Wei, Jiaxin Guo, Lizhi Lei, Ying Qin, Shimin Tao, Shiliang Sun, Yimeng Chen, and Liangyou Li. 2020a. Hw-tsc's participation at wmt 2020 quality estimation shared task. In *Proceedings of the Fifth Conference on Machine Translation, Volume 2: Shared Tasks Papers*, Online. Association for Computational Linguistics.

Zixuan Wang, Haijiang Wu, Xiaoli Wang, Xinjie Wen, Ruichen Wang, and Qingsong Ma. 2020b. Tencent submission for wmt20 quality estimation shared task. In *Proceedings of the Fifth Conference on Machine Translation, Volume 2: Shared Tasks Papers*, Online. Association for Computational Linguistics.

Evan J. Williams. 1959. *Regression Analysis*, volume 14. Wiley, New York, USA.

Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Lei Zhou, Liang Ding, and Koichi Takeda. 2020. Zero-shot translation quality estimation with explicit cross-lingual patterns. In *Proceedings of the Fifth Conference on Machine Translation, Volume 2: Shared Tasks Papers*, Online. Association for Computational Linguistics.

A Official Results of the WMT20 Quality Estimation Task 1

Tables 11, 12, 13, 14, 15, 16, 17 and 18 show the results for all language pairs and the multilingual variant, ranking participating systems best to worst using Pearson's r correlation as primary key for each of these cases.

Model	**Pearson** r	MAE	RMSE
TransQuest	**0.722**	0.480	0.596
Bergamot-LATTE (black-box)	0.718	**0.408**	**0.527**
TMUOU	0.686	0.418	0.543
IST and Unbabel (Kiwi-glass-box-ensemble)	0.673	0.433	0.569
IST and Unbabel (OpenKiwi-base)	0.583	0.547	0.719
WL Research	0.546	0.538	0.683
Bergamot-LATTE (glass-box)	0.489	0.895	1.062
jackielo ‡	0.462	0.918	1.141
BASELINE	0.376	0.788	0.999
NiuTrans †	0.732	0.529	0.653

Table 11: Official results of the WMT20 Quality Estimation Task 1 for the **Multilingual** variant. Baseline systems are highlighted in grey; † indicates teams that have been identified as having submitted more systems than the allowed limit to the leaderboard; ‡ indicates Codalab usernames of participants from whom we have not received further information.

Model	**Pearson** r	MAE	RMSE
• TransQuest	**0.554**	0.613	0.740
• Bergamot-LATTE (black-box)	0.544	**0.451**	**0.616**
IST and Unbabel (Kiwi-glass-box-ensemble)	0.523	0.470	0.635
Papago	0.498	0.454	0.637
TMUOU	0.482	0.455	0.625
Bergamot	0.476	0.483	0.636
XC	0.465	0.739	0.861
zhanghuimeng ‡	0.392	0.715	0.964
IST and Unbabel (OpenKiwi-base)	0.267	0.525	0.683
Bergamot-LATTE (glass-box)	0.259	0.819	0.940
WL Research	0.253	0.527	0.683
DexinWang ‡	0.246	0.503	0.680
Hancheng_Deng ‡	0.171	0.490	0.726
BASELINE	0.146	0.679	0.967
FVCRC	0.111	0.805	1.063
NiuTrans †	0.562	0.558	0.676

Table 12: Official results of the WMT20 Quality Estimation Task 1 for the **English-German** dataset. Teams marked with "•" are the winners, as they are not significantly outperformed by any other system according to the Williams Significance Test (Williams, 1959). Baseline systems are highlighted in grey; † indicates teams that have been identified as having submitted more systems than the allowed limit to the leaderboard; ‡ indicates Codalab usernames of participants from whom we have not received further information.

Model	**Pearson** r	MAE	RMSE
• TransQuest	**0.537**	0.675	0.831
Bergamot-LATTE (black-box)	0.530	**0.452**	**0.587**
IST and Unbabel (Kiwi-glass-box-ensemble)	0.494	0.459	0.592
XC	0.465	0.782	0.944
aj54 ‡	0.444	1.020	1.170
TMUOU	0.438	0.585	0.739
Bergamot	0.429	0.467	0.612
JXNU-CCLQ	0.426	0.709	0.890
IST and Unbabel (OpenKiwi-base)	0.346	0.518	0.684
Bergamot-LATTE (glass-box)	0.321	1.094	1.228
WL Research	0.298	0.796	0.970
RTM	0.259	68.010	68.414
BASELINE	0.190	0.885	1.068
FVCRC	0.085	0.873	1.059
NiuTrans †	0.551	0.499	0.654

Table 13: Official results of the WMT20 Quality Estimation Task 1 for the **English-Chinese** dataset. Teams marked with "•" are the winners, as they are not significantly outperformed by any other system according to the Williams Significance Test (Williams, 1959). Baseline systems are highlighted in grey; † indicates teams that have been identified as having submitted more systems than the allowed limit to the leaderboard; ‡ indicates Codalab usernames of participants from whom we have not received further information.

Model	**Pearson** r	MAE	RMSE
• TransQuest	**0.908**	0.300	0.392
• Bergamot-LATTE (black-box)	0.906	**0.281**	**0.388**
TMUOU	0.896	0.294	0.414
IST and Unbabel (Kiwi-glass-box-ensemble)	0.891	0.398	0.530
XC	0.882	0.556	0.661
Shrangin ‡	0.846	0.727	1.009
WL Research	0.821	0.393	0.520
Bergamot	0.796	0.438	0.554
IST and Unbabel (OpenKiwi-base)	0.708	0.508	0.655
RTM	0.703	0.517	0.654
Bergamot-LATTE (glass-box)	0.693	0.994	1.132
BASELINE	0.685	0.760	1.052
FVCRC	0.650	0.840	1.174
NiuTrans †	0.917	0.583	0.691

Table 14: Official results of the WMT20 Quality Estimation Task 1 for the **Romanian-English** dataset. Teams marked with "•" are the winners, as they are not significantly outperformed by any other system according to the Williams Significance Test (Williams, 1959). Baseline systems are highlighted in grey; † indicates teams that have been identified as having submitted more systems than the allowed limit to the leaderboard; ‡ indicates Codalab usernames of participants from whom we have not received further information.

Model	**Pearson** r	MAE	RMSE
• Bergamot-LATTE (black-box)	**0.826**	**0.427**	**0.540**
• TransQuest	0.824	0.485	0.604
TMUOU	0.792	0.493	0.636
IST and Unbabel (Kiwi-glass-box-ensemble)	0.770	0.740	0.919
XC	0.764	0.745	0.906
IST and Unbabel (OpenKiwi-base)	0.690	0.531	0.652
Bergamot	0.681	0.565	0.682
Bergamot-LATTE (glass-box)	0.642	0.918	1.096
WL Research	0.637	0.567	0.714
RTM	0.614	66.362	67.656
BASELINE	0.477	0.918	1.138
NiuTrans †	0.833	0.561	0.716

Table 15: Official results of the WMT20 Quality Estimation Task 1 for the **Estonian-English** dataset. Teams marked with "•" are the winners, as they are not significantly outperformed by any other system according to the Williams Significance Test (Williams, 1959). Baseline systems are highlighted in grey; † indicates teams that have been identified as having submitted more systems than the allowed limit to the leaderboard.

Model	**Pearson** r	MAE	RMSE
• TransQuest	**0.822**	0.372	**0.474**
Bergamot-LATTE (black-box)	0.814	**0.368**	0.475
IST and Unbabel (Kiwi-glass-box-ensemble)	0.792	0.433	0.549
TMUOU	0.785	0.397	0.511
XC	0.778	1.414	1.512
WL Research	0.687	0.452	0.594
Bergamot	0.662	0.486	0.612
IST and Unbabel (OpenKiwi-base)	0.604	0.497	0.648
Bergamot-LATTE (glass-box)	0.600	0.727	0.854
FVCRC	0.488	0.918	1.046
BASELINE	0.386	0.735	0.871
NiuTrans †	0.830	0.481	0.629

Table 16: Official results of the WMT20 Quality Estimation Task 1 for the **Nepalese-English** dataset. Teams marked with "•" are the winners, as they are not significantly outperformed by any other system according to the Williams Significance Test (Williams, 1959). Baseline systems are highlighted in grey; † indicates teams that have been identified as having submitted more systems than the allowed limit to the leaderboard.

Model	**Pearson** r	MAE	RMSE
• TransQuest	**0.685**	0.436	**0.534**
• Bergamot-LATTE (black-box)	0.682	**0.429**	0.539
TMUOU	0.668	0.459	0.572
IST and Unbabel (Kiwi-glass-box-ensemble)	0.639	0.506	0.642
XC	0.626	0.879	1.021
WL Research	0.577	0.492	0.614
IST and Unbabel (OpenKiwi-base)	0.565	0.515	0.634
Bergamot	0.560	0.490	0.602
RTM	0.541	49.675	50.774
Bergamot-LATTE (glass-box)	0.513	0.673	0.819
FVCRC	0.388	0.694	0.848
BASELINE	0.374	0.752	0.898
NiuTrans †	0.698	0.445	0.543

Table 17: Official results of the WMT20 Quality Estimation Task 1 for the **Sinhala-English** dataset. Teams marked with "•" are the winners, as they are not significantly outperformed by any other system according to the Williams Significance Test (Williams, 1959). Baseline systems are highlighted in grey, and † indicates teams have been identified as having submitted more systems than the allowed limit to the leaderboard.

Model	**Pearson** r	MAE	RMSE
• TransQuest	**0.808**	**0.402**	**0.583**
Bergamot-LATTE (black-box)	0.796	0.412	0.584
XC	0.784	0.603	0.759
TMUOU	0.781	0.433	0.622
IST and Unbabel (Kiwi-glass-box-ensemble)	0.767	0.428	0.613
WL Research	0.596	0.575	0.763
BASELINE	0.548	0.825	1.193
Mak	0.543	0.590	0.811
jackielo ‡	0.411	0.878	1.267
FVCRC	0.400	0.831	1.220
NiuTrans †	0.816	0.535	0.687

Table 18: Official results of the WMT20 Quality Estimation Task 1 for the **Russian-English** dataset. Teams marked with "•" are the winners, as they are not significantly outperformed by any other system according to the Williams Significance Test (Williams, 1959). Baseline systems are highlighted in grey; † indicates teams that have been identified as having submitted more systems than the allowed limit to the leaderboard; ‡ indicates Codalab usernames of participants from whom we have not received further information.

B Official Results of the WMT20 Quality Estimation Task 2 (Sentence-level)

Tables 19 and 20 show the evaluation results for English-German and English-Chinese respectively, ranking participating systems best to worst using Pearson's r correlation as primary key for each language pair.

Model	**Pearson** r	MAE	RMSE
• HW-TSC	0.758	0.099	0.133
Bering Lab	0.723	0.107	0.140
IST and Unbabel (Kiwi-glass-box)	0.633	0.137	0.178
NJUNLP	0.618	0.129	0.160
NICT Kyoto	0.615	0.151	0.197
Bergamot	0.613	0.130	0.160
IST and Unbabel (OpenKiwi-base)	0.531	0.138	0.180
TransQuest	0.499	0.149	0.184
zhanghuimeng ‡	0.494	0.163	0.198
DexinWang ‡	0.402	0.155	0.196
BASELINE	0.392	0.150	0.190
NiuTrans †	0.649	0.123	0.154

Table 19: Official results of the WMT20 Quality Estimation Task 2 sentence-level for the **English-German** dataset. Teams marked with "•" are the winners, as they are not significantly outperformed by any other system according to the Williams Significance Test (Williams, 1959). Baseline systems are highlighted in grey; † indicates teams that have been identified as having submitted more systems than the allowed limit to the leaderboard; ‡ indicates Codalab usernames of participants from whom we have not received further information.

Model	**Pearson** r	MAE	RMSE
• Tencent Inc.	0.664	0.129	0.160
• IST and Unbabel (Kiwi-glass-box)	0.651	0.135	0.171
NICT Kyoto	0.643	0.129	0.161
NJUNLP	0.642	0.129	0.161
Bergamot	0.613	0.136	0.169
TransQuest	0.612	0.135	0.168
IST and Unbabel (OpenKiwi-base)	0.593	0.143	0.175
niuniuniu ‡	0.569	0.142	0.177
aj54 ‡	0.552	0.145	0.176
BASELINE	0.506	0.147	0.181
NiuTrans †	0.675	0.125	0.156

Table 20: Official results of the WMT20 Quality Estimation Task 2 sentence-level for the **English-Chinese** dataset. Teams marked with "•" are the winners, as they are not significantly outperformed by any other system according to the Williams Significance Test (Williams, 1959). Baseline systems are highlighted in grey; † indicates teams that have been identified as having submitted more systems than the allowed limit to the leaderboard; ‡ indicates Codalab usernames of participants from whom we have not received further information.

Findings of the WMT 2020 Shared Tasks in Unsupervised MT and Very Low Resource Supervised MT

Alexander Fraser

Center for Information and Language Processing, LMU Munich, Germany
fraser@cis.lmu.de

Abstract

We describe the WMT 2020 Shared Tasks in Unsupervised MT and Very Low Resource Supervised MT. In both tasks, the community studied German↔Upper Sorbian MT, which is a very realistic machine translation scenario (unlike the simulated scenarios used in particular in much of the unsupervised MT work in the past). We were able to obtain most of the digital data available for Upper Sorbian, a minority language of Germany, which was the original motivation for the Unsupervised MT shared task. As we were defining the task, we also obtained a small amount of parallel data (about 60000 parallel sentences), allowing us to offer a Very Low Supervised MT task as well. Six primary systems participated in the unsupervised shared task, two of these systems used additional data beyond the data released by the organizers. Ten primary systems participated in the very low resource supervised task. The paper discusses the background, presents the tasks and results, and discusses best practices for the future.

1 Introduction

There is no machine translation available for most of the approximately 7000 languages spoken on the planet Earth. This is because very limited or no parallel corpora are available. Research on unsupervised and very low resource machine translation is important for alleviating this problem. Unsupervised machine translation requires only monolingual data, while very low resource supervised machine translation uses very limited parallel data.

At WMT 2018 and WMT 2019, the first shared task (Bojar et al., 2018) and second shared task (Barrault et al., 2019) on Unsupervised Machine Translation (UMT), were held as part of the news translation track. In both 2018 and 2019 the scenarios simulated low resource setups using medium or high resource languages. In 2018, the language pairs were Turkish-English, Estonian-English and German-English. In 2019, we tested unsupervised systems for German to Czech unsupervised translation (where no German/Czech parallel data was allowed).

In the 2020 shared task we proposed a third edition on UMT, which aimed at a more realistic scenario, German to Upper Sorbian (and Upper Sorbian to German) translation. Upper Sorbian is a minority language of Germany that is in the Slavic language family (e.g., related to Lower Sorbian, Czech and Polish), and we provide here most of the digital data that is available for Upper Sorbian, as far as we know. The amount of monolingual data available for Upper Sorbian is quite small (see below), making unsupervised machine translation very challenging.

While working on the UMT task we were able to obtain a very small amount of parallel data (about 60000 parallel sentences) for this language pair, which allowed us to additionally offer the very low resource supervised translation task.

The exact tasks studied in the shared tasks were:

- Unsupervised Machine Translation: German to Upper Sorbian. Upper Sorbian to German.
- Very Low Resource Supervised Machine Translation: German to Upper Sorbian. Upper Sorbian to German.

The data used can be downloaded from the shared task webpage.[1]

This paper will give some background for the task, discuss the data made available (with a particular focus on considerations for benchmarks for future UMT and very low resource MT research), discuss how the tasks went with a presentation of the results, and then conclude.

[1] http://www.statmt.org/wmt20/unsup_and_very_low_res/

Proceedings of the 5th Conference on Machine Translation (WMT), pages 765–771
Online, November 19–20, 2020.

2 Background

The 2020 shared tasks focused on the very low resource language Upper Sorbian (a Slavic minority language spoken in the Eastern part of Germany). The Sorbian language (referring to Upper Sorbian and Lower Sorbian) has a special protected status in the "Grundgesetz" of Germany (similar to a constitution).

Working with the Sorbian Institute[2] we initially prepared an unsupervised MT task. We expect this task to become a standard benchmark for unsupervised MT development. This task particularly relies on having high quality Upper Sorbian monolingual data, which we obtained from the Sorbian Institute and the Witaj Sprachzentrum. We also offered data which LMU Munich obtained through web crawling. Finally, the Sorbian Institute has also provided medium quality data which has not been quality checked.

The Witaj Sprachzentrum (Witaj Language Center[3]) then provided a small training corpus of German/Upper Sorbian parallel data, which was used in the Very Low Resource Supervised Machine Translation task. We expect this task to become a standard task for very low resource scenarios, and note that the results are important as they will directly inform efforts to create state-of-the-art machine translation systems for use by the Upper Sorbian community.

The Witaj Sprachzentrum provided development and "development test" (not blind test) sets for German to Upper Sorbian and Upper Sorbian to German translation. The development set is used to tune parameters, while the devtest set is used to measure progress using automatic metrics. The Witaj Sprachzentrum also provided the blind test sets, which were released just in time for the evaluation.

We note that CIS (LMU Munich), the Sorbian Institute and the Witaj Sprachzentrum hope to organize a task for Unsupervised Lower Sorbian translation in the near future.

LMU Munich would be interested in adding new typologically diverse languages as unsupervised and/or very low resource tasks in future WMT Shared Tasks.

[2] https://www.serbski-institut.de/en/Institute/

[3] https://www.witaj-sprachzentrum.de/

3 Basic Tasks and Evaluation

We provide some basic details about the tasks and evaluation here.

Unsupervised MT: Unsupervised translation from German to Upper Sorbian, and unsupervised translation from Upper Sorbian to German. We allow the use of all German data released for WMT, except that the German side of the small parallel German/Upper Sorbian training corpus may not be used. All Upper Sorbian data we release may be used. No other language data may be used.

Very Low Resource Supervised MT: Supervised translation from German to Upper Sorbian, and supervised translation from Upper Sorbian to German. We allow the use of all German and Upper Sorbian data released for WMT, including the small parallel German/Upper Sorbian training corpus. Other WMT data for other languages may also be used. Upper Sorbian is a Slavic language which has strong similarities to Czech, so the German/Czech data we discuss below may be of particular interest for multilingual systems.

We used automatic metrics for the evaluation of this task. We believe that manual evaluation may not be so necessary for unsupervised MT and very low resource MT development, because automatic metrics worked well at relatively low translation quality levels in the past. Translation to Upper Sorbian would have been fairly difficult to evaluate in a human evaluation, as we do not have easy access to a large number of native speakers.

4 Data

We describe the data released for the two scenarios, Unsupervised and Very Low Resource Supervised.

The Unsupervised MT scenario allowed the use of all German data released for WMT, except that the German side of the small parallel German/Upper Sorbian training corpus could not be used. All Upper Sorbian data available on the web page was allowed. No other language data could be used (no parallel, no other monolingual data sets for any language except those explicitly listed as usable for Unsupervised).

The Very Low Resource Supervised MT scenario allowed the use of all German and Upper Sorbian data released for WMT, including the 60000 sentence parallel German/Upper Sorbian training corpus. Other WMT 2020 data for other languages were also allowed. Upper Sorbian is a Slavic language which is related to Czech, so the Ger-

man/Czech parallel data we briefly describe next were of particular interest for building multilingual systems. We would also like to express our thanks to the Opus project for the German/Czech parallel data.

In addition to training data, we also provided **development data**. Specifically, we provided development and test sets for German to Upper Sorbian and Upper Sorbian to German. These were usable for both Unsupervised and Very Low Resource. The dev set was intended for use for parameter tuning (and the participants were asked to not use it as a parallel training corpus). The test set was intended for system evaluation during development (likewise, the participants were asked to not use it as a parallel training corpus). We would like to thank Jindřich Libovický for creating the data splits and for work on the training data.

The specific data sets are presented in Table 1.

The translation output was submitted as real case, detokenized, and in SGML format. We used the Matrix for submission, thanks to Barry Haddow for assistance. The standard script wrap-xml.perl was used to create the sgm files for upload.

5 Results

In this section we discuss the results for the Unsupervised track, a non-constraint system that does not use parallel German/Sorbian data but does use German-English parallel data (which does not meet the constraints for the Unsupervised Task), and the Very Low Resource task.

We note that the absolute BLEU scores are surprisingly high, this may be due to unusually homogeneous train and test sets.

5.1 Unsupervised MT

There were four submissions to the Unsupervised MT which used only the allowed data for the track, see Table 2. The citations are listed in the table. Highlights of systems here included the use of transfer learning to obtain better initialization for the lower resource language and refinements of BPE, see the papers (and the analysis below) for further details.

5.2 Unsupervised MT with Multilingual Transfer Learning

Two primary submissions to Unsupervised MT were not restricted to the allowed data for the track, see Table 3. One system used data mined from German and Upper Sorbian Wikipedia, see (Dutta et al., 2020) for more details. The creators of the other system (Li et al., 2020) used parallel data for English to German to initialize an unsupervised system, which is a reasonable scenario to consider. This was highly effective. We suggest that this result be used as an initial benchmark for multilingual transfer-based unsupervised systems. We consider these results to be separate from the simpler unsupervised benchmark that was previously proposed. See the subsequent analysis and discussion for more details.

5.3 Very Low Resource MT

A Moses baseline from the Witaj Sprachzentrum using only the data from the shared task web page scored 46.36 for DE-HSB (BLEU-cased, 11b) and 47.70 for HSB-DE (BLEU-cased, 11b). The results for the shared task systems are presented in Table 4. Note that the last system did not submit a shared task paper.

6 Analysis

Overall we proposed two new benchmark tasks for low resource MT. The Unsupervised MT track can be used as a benchmark for testing new unsupervised MT approaches. The Very Low Resource track can be used as a benchmark for testing MT systems when one language has very little parallel data. Additionally, one system which was conceptually unsupervised (in that it used no parallel German-Upper Sorbian data), effectively defined a new benchmark, which we will call Unsupervised MT with Multilingual Transfer, by using additional English-German parallel data, see below.

6.1 Task Definitions

Unsupervised MT: The data released for the Unsupervised MT track should be considered to be a new realistic benchmark for testing unsupervised MT systems. Artetxe et al. (2020) (a best practices paper at ACL 2020) made a number of suggestions in terms of how to setup future Unsupervised MT research. The guidelines we set agree with their guidelines except with respect to one point. We disagree with Artetxe et al. in terms of whether development sets should be made available. The dev set is traditionally used for setting a small number of open parameters. Artetxe et al. argue that this data should not be made available, as it represent parallel data that is not really in spirit with

Monolingual Upper Sorbian Data	
sorbian_institute_monolingual.hsb.gz	Upper Sorbian monolingual data provided by the Sorbian Institute. This contains a high quality corpus and some medium quality data which were mixed together.
witaj_monolingual.hsb.gz	Upper Sorbian monolingual data provided by the Witaj Sprachzentrum (high quality).
web_monolingual.hsb.gz	Upper Sorbian monolingual data scraped from the web by CIS, LMU (thanks to Alina Fastowski). This should be used with caution, it is probably noisy, it might erroneously contain some data from related languages.
Monolingual German Data	
	See the news translation task web page for allowed monolingual German data. All monolingual German sets allowed in this years News task were allowed in both scenarios.
Upper Sorbian side of parallel training corpus	
train.hsb-de.hsb.gz	This file was usable for both the Unsupervised and Very Low Resource Supervised scenarios. In the Unsupervised scenario, it was used as a small high quality monolingual corpus.
German side of parallel training corpus	
train.hsb-de.de.gz	This file was not usable for Unsupervised.
Dev and Test Sets	
devtest.tar.gz	The participants were requested to please use dev to tune system parameters, and test to measure progress. These files were allowed in both tracks.
German/Czech Parallel Data	
	This data was not allowed for Unsupervised. For Very Low Resource MT we allowed all German/Czech parallel corpora obtainable from the Opus project. The de-cs corpora we particularly recommended using are: Europarl v8 and JW300 v1. These two corpora may be somewhat similar to the DE-HSB parallel training and test data, but this is far from certain.

Table 1: Data sets for both tasks.

System Name	DE-HSB	HSB-DE	citation
LMU Munich-NMT	35.0	31.6	(Chronopoulou et al., 2020)
CUNI-Synthetic	25.0	23.4	(Kvapilíková et al., 2020)
NITS-CNLP	15.4		(Singh et al., 2020)
rug_hsbde_unsup_sel		20.1	(Edman et al., 2020)

Table 2: Four primary submissions to the Unsupervised MT Task using only the allowed data for the track, sorted by DE-HSB BLEU score.

System Name	DE-HSB	HSB-DE	citation
SJTU-NICT	40.3	32.8	(Li et al., 2020)
UdS-DFKI	10.3	9.0	(Dutta et al., 2020)

Table 3: Two primary submissions to the Unsupervised MT Task using additional English-German parallel data (row 1) and German and Upper Sorbian monolingual data from Wikipedia (row 2).

the unsupervised MT paradigm. In practice one could create a "very very low resource" system by simply training a supervised system on this data. We however argue that system developers need to have access to standard dev and devtest sets in order to measure progress. Having access to a dev set simulates having access to, e.g., a native speaker of the low resource language, who is able to look at outputs during MT development and judge the changes in quality obtained through simple hyperparameter changes. In our view the use of a devtest set to measure BLEU (which is completely standard in all unsupervised MT research) is similar to this, and of course also represents a small amount of available parallel data. A further argument for our view is that having a designated dev set for parameter tuning strongly helps with replicability of results. However, it is important in our view that this data not be trained on (in terms of using it as a parallel corpus to training a supervised system). The restriction of the rest of the data to be monolingual and carefully controlled makes sense for the straightforward testing of bilingual unsupervised systems, but we describe an interesting different multilingual scenario next.

Unsupervised MT with Multilingual Transfer: SJTU-NICT submitted a system to Unsupervised MT which looked at the reasonable scenario of assuming that the high resource language (German for German-Upper Sorbian) has parallel corpora with another language. In their system they used an English-German parallel corpus to initialize their German-Upper Sorbian Unsupervised MT system. This makes their system conceptually a multilingual system. Another obvious choice would be to try German-Czech and/or German paired with another Slavic language (and in fact, in the Very Low Resource Track, this was a common strategy). There were no competing systems for this unplanned benchmark (which was essentially created by SJTU-NICT's submission), but we suggest that this result is also interesting for comparison in future research and intend to offer a new track for this scenario in the shared task next year.

Very Low Resource MT: Please see the shared task system descriptions to see the wide variation in the exact data used for the Very Low Resource task.

There was obviously a high level of interest overall in this task, and we already have plans to offer more tasks of this kind as well. This task is very interesting as even if an unsupervised MT system is actually deployed for a particular language pair, the users of the system can post-edit the output of that system and will soon therefore be interested in training higher quality supervised systems using the Very Low Resource scenario. The parallel data used in this track was in fact created manually by the Witaj Sprachzentrum in order to be used by the Upper Sorbian community in Germany (to train the baseline Moses system we mentioned in the results section, and for future neural machine translation work which will be informed by the results produced by the research community). As the result of the existence of this pipeline more data may become available in the future.

6.2 What Worked Well

We highlight four interesting trends in terms of the results. Two of the trends we observe here involve transfer learning (i.e., pretraining). For all three

System Name	DE-HSB	HSB-DE	citation
SJTU-NICT	60.7	58.5	(Li et al., 2020)
Helsinki-NLP	57.9	59.6	(Scherrer et al., 2020)
NRC-CNRC	57.3	58.9	(Knowles et al., 2020)
LMU-supervised-ensemble	56.5	57.6	(Libovický et al., 2020)
CUNI-Transfer	55.5	56.9	(Kvapilíková et al., 2020)
Brown-NLP-b	46.2	45.7	(Berckmann and Hiziroglu, 2020)
IITBHU-NLPRL-DE-HSB	45.9	47.9	(Baruah et al., 2020)
Adobe-AMPS	45.2	47.6	(Singh, 2020)
UdS-DFKI	40.9		(Dutta et al., 2020)
HierarchicalTransformer	38.2	40.1	

Table 4: Ten primary systems submitted to the Very Low Resource Task, sorted by DE-HSB BLEU score.

benchmarks (Unsupervised, Unsupervised with Multilingual Transfer, and Very Low Resource) transfer learning is a critical component of many or all systems. The trends we will discuss are transfer learning using monolingual corpora and transfer learning using bilingual corpora. For Unsupervised, transfer learning using monolingual corpora was an important aspect of successful systems. Please see the system description papers for more detail about how the monolingual data was used to initialize successful Unsupervised systems. For Unsupervised with Multilingual Transfer, the use of English-German bilingual corpora to initialize the Unsupervised system seems to have been highly effective, with a particularly strong result for the DE-HSB translation direction, perhaps due to a very strong starting point for the German encoder. For Very Low Resource, heavy usage of both types of transfer were made as well, with one particular focus being on efforts to leverage the similarity of Czech and Upper Sorbian using Czech/German parallel corpora.

The third trend was that another area of study for many systems was word segmentation (typically with a variant of BPE) and/or the use of morphological information, sometimes learned in translation at BPE or character level, sometimes monolingually. Finally the fourth trend was that there was also a significant focus on trying to make Czech more like Upper Sorbian using a variety of techniques, and/or sampling German data like Upper Sorbian data.

7 Conclusion

The WMT 2020 Shared Tasks in Unsupervised MT and Very Low Resource Supervised MT have created three interesting new benchmarks for future research. In addition to the initially proposed benchmark on straight-forward bilingual Unsupervised MT, we suggest the use of the Unsupervised Multilingual Transfer result as an additional new benchmark. The very low resource MT benchmark also generated strong participation from the research community.

We hope that additionally we have played a role in raising the interest of the NLP community in Upper Sorbian language processing, and that this interest will extend to Lower Sorbian language processing and in fact extend to working on other understudied languages as well.

We plan to organize similar tasks for the three benchmarks next year. Options include, e.g., more parallel data for Upper Sorbian, an Unsupervised Task for Lower Sorbian, and more ambitiously the inclusion of new typologically diverse languages. Please don't hesitate to contact us if you are interested, particularly if you have access to appropriate data.

8 Acknowledgments

This work has received funding from the European Research Council (ERC) under grant agreement No. 640550. Thanks very much to coorganizers: Hauke Bartels - Sorbian Institute, Olaf Langner - Witaj Sprachzentrum, Marcin Szczepanski - Sorbian Institute.

References

Mikel Artetxe, Sebastian Ruder, Dani Yogatama, Gorka Labaka, and Eneko Agirre. 2020. A call for more rigor in unsupervised cross-lingual learning. In *the 58th Annual Meeting of the Association for Computational Linguistics*.

Loïc Barrault, Ondrej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. 2019. Findings of the 2019 Conference on Machine Translation (WMT19). In *the Fourth Conference on Machine Translation, WMT 2019, Florence, Italy, August 1-2, 2019 - Volume 2: Shared Task Papers, Day 1*.

Rupjyoti Baruah, Rajesh Kumar Mundotiya, Amit Kumar, and Anil Kumar Singh. 2020. NLPRL system for very low resource supervised machine translation. In *the Fifth Conference on Machine Translation*.

Tucker Berckmann and Berkan Hiziroglu. 2020. Low resource translation as language modeling. In *the Fifth Conference on Machine Translation*.

Ondřej Bojar, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, and Christof Monz. 2018. Findings of the 2018 Conference on Machine Translation (WMT18). In *the Third Conference on Machine Translation, Volume 2: Shared Task Papers*.

Alexandra Chronopoulou, Dario Stojanovski, Viktor Hangya, and Alexander Fraser. 2020. The LMU Munich system for the WMT 2020 unsupervised machine translation shared task. In *the Fifth Conference on Machine Translation*.

Sourav Dutta, Jesujoba O. Alabi, Saptarashmi Bandyopadhyay, Dana Ruiter, and Josef van Genabith. 2020. UdS-DFKI@WMT20: Unsupervised MT and Very Low Resource Supervised MT for German - Upper Sorbian. In *the Fifth Conference on Machine Translation*.

Lukas Edman, Antonio Toral, and Gertjan van Noord. 2020. Data selection for unsupervised translation of German–Upper Sorbian. In *the Fifth Conference on Machine Translation*.

Rebecca Knowles, Samuel Larkin, Darlene Stewart, and Patrick Littell. 2020. NRC systems for low resource German-Upper Sorbian machine translation 2020: Transfer learning with lexical modifications. In *the Fifth Conference on Machine Translation*.

Ivana Kvapilíková, Tom Kocmi, and Ondřej Bojar. 2020. CUNI systems for the unsupervised and very low resource translation task in WMT20. In *the Fifth Conference on Machine Translation*.

Zuchao Li, Hai Zhao, Rui Wang, Kehai Chen, Masao Utiyama, and Eiichiro Sumita. 2020. SJTU-NICT's supervised and unsupervised neural machine translation systems for the WMT20 news translation task. In *the Fifth Conference on Machine Translation*.

Jindřich Libovický, Viktor Hangya, Helmut Schmid, and Alexander Fraser. 2020. The LMU Munich system for the WMT20 very low resource supervised MT task. In *the Fifth Conference on Machine Translation*.

Yves Scherrer, Stig-Arne Grönroos, and Sami Virpioja. 2020. The University of Helsinki and Aalto University submissions to the WMT 2020 news and low-resource translation tasks. In *the Fifth Conference on Machine Translation*.

Keshaw Singh. 2020. Adobe AMPS's submission for very low resource supervised translation task at WMT20. In *the Fifth Conference on Machine Translation*.

Salam Michael Singh, Thoudam Doren Singh, and Sivaji Bandyopadhyay. 2020. The NITS-CNLP system for the unsupervised machine translation task at WMT 2020. In *the Fifth Conference on Machine Translation*.

Cross-Lingual Transformers for Neural Automatic Post-Editing

Dongjun Lee
Bering Lab, Republic of Korea
djlee@beringlab.com

Abstract

In this paper, we describe the Bering Lab's submission to the WMT 2020 Shared Task on Automatic Post-Editing (APE). First, we propose a cross-lingual Transformer architecture that takes a concatenation of a source sentence and a machine-translated (MT) sentence as an input to generate the post-edited (PE) output. For further improvement, we mask incorrect or missing words in the PE output based on word-level quality estimation and then predict the actual word for each mask based on the fine-tuned cross-lingual language model (XLM-RoBERTa). Finally, to address the over-correction problem, we select the final output among the PE outputs and the original MT sentence based on a sentence-level quality estimation. When evaluated on the WMT 2020 English-German APE test dataset, our system improves the NMT output by −3.95 and +4.50 in terms of TER and BLEU, respectively.

1 Introduction

Automatic post-editing (APE) is the task of automatically correcting errors in the output of a machine translation (MT) system by learning from human corrections (Chatterjee et al., 2019). APE can be viewed as a cross-lingual sequence-to-sequence task, which takes a source sentence and the corresponding MT output as inputs and generates the post-edited (PE) output.

Our work is inspired by XLM-RoBERTa (XLM-R) (Conneau et al., 2019), a cross-lingual language model, which shows the state-of-the-art performance for a wide range of cross-lingual tasks. XLM-R takes a concatenation of two sentences in different languages as an input to generate cross-lingual representations. Similarly, we propose a Transformer (Vaswani et al., 2017) architecture for APE in which the encoder uses the same architecture as XLM-R.

In addition, we use XLM-R-based translation quality estimation (QE) (Lee, 2020) to further improve the PE output of the Transformer. QE is the task of estimating the quality of the MT output when only the source text is provided (Fonseca et al., 2019). We use two granularity levels of QE: word-level and sentence-level. Based on the word-level QE, we try to correct the wrong words or insert the missing words in the PE output. Through the sentence-level QE, we select the best translation among PE outputs and the original MT sentence to prevent over-correction (i.e., one of the APE models rephrases an already correct MT output).

Our contributions are summarized as follows:

- We propose a Transformer (Vaswani et al., 2017) architecture for APE in which an encoder takes concatenation of a source and MT sentence as an input to generate a cross-lingual representation and a decoder generates a PE output.
- We incorporate a word-level QE-based word masking. We replace `BAD` words with `<mask>` token or insert `<mask>` token for missing words in the PE output of the Transformer based on word-level QE.
- To predict the most probable word for each masked token, we use XLM-R (Conneau et al., 2019) that is fine-tuned using the translation language modeling (TLM) objective (Conneau and Lample, 2019).
- To address the over-correction problem, we introduce a sentence-level QE-based output selection. We select the sentence with the lowest predicted HTER among the MT and PE sentences as the final output.

In the experiment using the WMT 2020 English-German APE test set, our system achieves −3.95

Proceedings of the 5th Conference on Machine Translation (WMT), pages 772–776

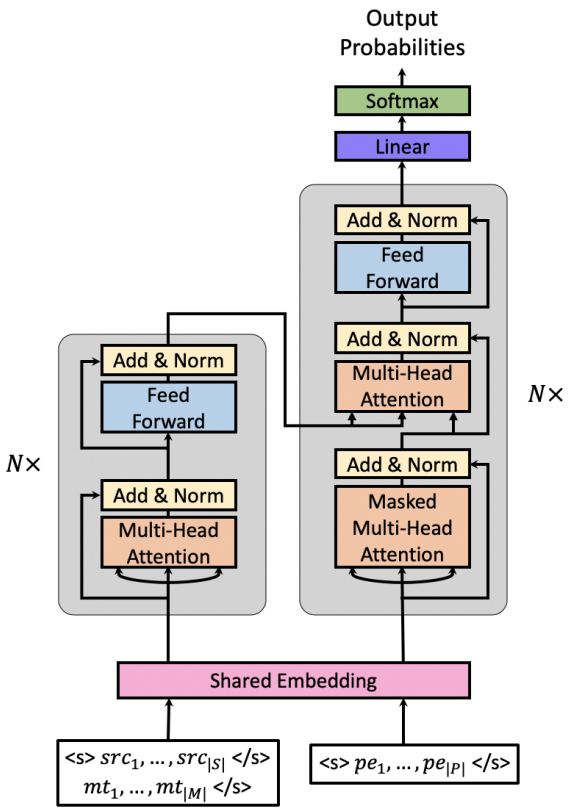


Figure 1: The cross-lingual Transformer architecture for APE.

TER and +4.50 BLEU improvement over the baseline (NMT output).

2 Methodology

Our approach for APE comprises three components: 1) a cross-lingual Transformer, 2) word masking based on word-level quality estimation (QE) and XLM-R-based mask prediction, and 3) output selection based on sentence-level QE.

2.1 Cross-Lingual Transformer for APE

As the first step of APE, we propose a cross-lingual Transformer (Vaswani et al., 2017) architecture that takes the concatenation of a source and MT sentence as a single input and generates a post-edited (PE) sentence, as illustrated in Figure 1.

A source sentence and its corresponding MT sentence are tokenized based on the same BPE model (Sennrich et al., 2016) that is trained using shared vocabulary of English and German. The input of the Transformer is a concatenated sequence of source tokens and MT tokens along with special tokens (<s>, </s>) as follows:

<s> src_1, ..., $src_{|S|}$ </s> mt_1, ..., $mt_{|M|}$ </s>

The output of the Transformer is a sequence of PE tokens that is also tokenized based on the same BPE model. Since the input and output use the shared dictionary, we tie the weights of the encoder word embedding layer, decoder word embedding layer, and decoder output layer. The rest of the model architecture follows that of Vaswani et al. (2017).

2.2 Word-level QE-based Word Masking and XLM-R-based Mask Prediction

We further improve the APE performance based on the word-level quality estimation (QE) (Fonseca et al., 2019) and XLM-R-based mask prediction (Conneau et al., 2019).

Word-QE-based Masking We use the word-level QE to predict if a word in the MT sentence is OK or BAD and if there are any missing words. We replace the words predicted as BAD with the <mask> token and insert the <mask> token where the missing words are predicted to exist. For the word-level QE, we use the same model architecture and hyperparameters from Lee (2020) but with the probability threshold for BAD as 0.8 instead of 0.5 because masking the correct token may degrade APE performance.

XLM-R Fine-Tuning We fine-tune pre-trained XLM-R using a parallel corpus based on the translation language modeling (TLM) objective (Conneau and Lample, 2019). A source (English) and target (German) sentences are tokenized with the same BPE model (Sennrich et al., 2016), which is trained based on shared vocabulary. We concatenate source and target tokens with a separation token (</s>) and use it as an input of XLM-R. Then, we randomly mask 20% of the BPE tokens in the target sentences and train the model to correctly predict the masked tokens.

Mask Prediction We use the concatenated sequence of source tokens and masked MT tokens as the input to the fine-tuned XLM-R. To predict the corresponding word for each masked token, we follow the highest probability first strategy proposed by Lawrence et al. (2019). We replace the <mask> tokens iteratively, and in each step, the <mask> token predicting the word with the highest probability is replaced with the predicted word.

2.3 Sentence-level QE-based Output Selection

There are cases where the APE models can degrade translation quality owing to unnecessary corrections, known as the over-correction problem (Fonseca et al., 2019). To prevent this, we select the best translation among the MT sentence and output sentences from APE models based on a sentence-level QE.

Sentence-level QE aims to predict the human translation error rate (HTER) (Snover et al., 2006) of the MT sentence, which measures the required amount of human editing to fix the MT sentence. We use the XLM-R-based sentence-level QE model proposed by Lee (2020) to predict the HTER for each of 1) the MT sentence, 2) PE output sentence of the cross-lingual Transformer, and 3) PE output sentence of word-level QE-based mask prediction. Finally, we select the sentence with the lowest predicted HTER as the final PE output.

2.4 Data Augmentation

Supervised learning for APE requires triplets comprised of source sentences, machine-translated (MT) sentences, and human post-edited (PE) sentences. Since the cost involved in achieving PE sentences is significant, we use a parallel corpus including only source and target sentences to build artificial triplets following the ideas from Negri et al. (2018).

First, we split the parallel corpus into a training set and test set. We then train an NMT model with the training set and use the test set to generate artificial triplets. We generate MT sentences based on the trained NMT model and we use the target sentences of the parallel corpus as PE sentences. We repeat this process with different data splits to amass large quantities of artificial triplets. Finally, we oversample the human-labeled triplets and merge them with the artificially-generated triplets to build a final training dataset (Junczys-Dowmunt and Grundkiewicz, 2018).

3 Experiments

3.1 Experimental Setup

We evaluate our model with the WMT 2020 English-German APE dataset.[1] For the evaluation metrics, we use the translation error rate (TER) (Snover et al., 2006) and BLEU (Papineni et al., 2002).

To generate artificial triplets (§2.4), we use the English-German parallel corpus provided by the shared task that consists of 23,440,059 pairs. We use 90% of the pairs to train a Transformer (Vaswani et al., 2017) NMT model using OpenNMT-py (Klein et al., 2017) and the rest of the pairs to generate artificial triplets. As a result of running the process five times with different data splits, we achieve 11,720,029 artificial triplets.

As a final training dataset, we oversample the official English-German APE dataset that consists of 7000 triplets 50 times and merge them with artificial triplets. We use the final triplets to train the cross-lingual Transformer (§2.1) and source-PE pairs to train the XLM-R with a TLM objective (§2.2).

3.2 Model Configuration

For the cross-lingual Transformer, we follow most of the hyperparameters from the base model of Vaswani et al. (2017), but for 5 epochs with early stopping. For the ensembling, we train five models with different random seeds. For the word-level and sentence-level quality estimation, we follow the model architectures and hyperparameters from Lee (2020). For mask prediction, we fine-tune XLM-R-Large (Conneau et al., 2019) using the Adam optimizer (Kingma and Ba, 2014) with a learning rate of 5e-6, a batch size of 8 for 1 epoch, and a dropout (Hinton et al., 2012) rate of 0.1.

3.3 Experimental Result

Table 1 presents the result of the ablation analysis for the proposed methods without the ensemble on the dev set. First, our cross-lingual Transformer improves the MT output by -1.85 TER and $+2.33$ BLEU. Sentence-level QE-based output selection further improves the performance of -0.42 TER and $+0.29$ BLEU. This demonstrates that our sentence-level QE-based output selection is effective for addressing the over-correction problem. Alternatively, when we use the word-level QE-based mask prediction model instead of the cross-lingual Transformer, the TER and BLEU are improved over the baseline by -1.10 and $+0.62$, respectively. This result shows that our word-masking and mask prediction models also significantly improve the translation quality. When we add the mask prediction model after the cross-lingual Transformer, the TER is improved by -0.27, but the BLEU slightly

[1] http://www.statmt.org/wmt20/ape-task.html

Systems	TER↓	BLEU↑
Baseline (MT Output)	31.37	50.37
APE Transformer	29.52	52.70
APE Transformer + Sentence-QE	29.10	52.99
Word-QE + Sentence-QE	30.27	50.83
APE Transformer + Word-QE + Sentence-QE	28.83	52.80
+ Ensemble	**28.47**	**53.82**

Table 1: Ablation analysis without ensemble on the WMT 2020 English-German APE *dev* dataset.

Systems	TER↓	BLEU↑
HW-TSC	**20.21**	**66.89**
MinD	26.99	55.77
POSTECH-ETRI	27.02	56.37
Ours - Primary (Bering Lab)	27.61	54.71
Ours - Contrastive (Bering Lab)	27.96	54.60
POSTECH	28.22	54.51
Baseline (MT output)	31.56	50.21
KAISTxPAPAGO	32.00	49.21

Table 2: Official results evaluated on the WMT 2020 English-German APE *test* dataset.

decreased (-0.19). Finally, through the ensemble, we achieve an additional performance gain of -0.36 and $+1.02$ for the TER and BLEU, respectively.

Table 2 presents the official result evaluated on the WMT 2020 English-German APE test set. Our primary system contains all of the proposed methods, whereas the contrastive system does not contain word-level QE-based mask prediction. As can be seen, our primary system outperformed the contrastive system in terms of both TER and BLEU. In addition, our primary system achieves -3.95 TER and $+4.50$ BLEU improvement over the NMT output.

4 Conclusion

In this paper, the Bering Lab's submission to the WMT 2020 English-German APE shared task is described. A cross-lingual Transformer architecture is proposed for APE in which a single encoder takes the concatenation of a source and a MT sentences as an input to generate intermediary cross-lingual representations, and then a decoder outputs post-edited results. In addition, methods to improve the APE performance through translation QE are proposed. First, the incorrect or missing words in the post-edited output are masked based on a word-level QE. Then, the actual word for each mask is predicted based on the fine-tuned XLM-R using the translation language modeling (TLM) objective. Finally, a sentence-level QE-based output selection method is proposed to prevent over-correction. The experimental results show that our APE system significantly improves the NMT output in terms of both TER and BLEU.

References

Rajen Chatterjee, Christian Federmann, Matteo Negri, and Marco Turchi. 2019. Findings of the wmt 2019 shared task on automatic post-editing. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 11–28.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems*, pages 7057–7067.

Erick Fonseca, Lisa Yankovskaya, André F. T. Martins, Mark Fishel, and Christian Federmann. 2019. Find-

ings of the WMT 2019 shared tasks on quality estimation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 1–10, Florence, Italy. Association for Computational Linguistics.

Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.

Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2018. MS-UEdin submission to the WMT2018 APE shared task: Dual-source transformer for automatic post-editing. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 822–826, Belgium, Brussels. Association for Computational Linguistics.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proc. ACL*.

Carolin Lawrence, Bhushan Kotnis, and Mathias Niepert. 2019. Attending to future tokens for bidirectional sequence generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1–10.

Dongjun Lee. 2020. Two-phase cross-lingual language model fine-tuning for machine translation quality estimation. In *Not published yet.*

Matteo Negri, Marco Turchi, Rajen Chatterjee, and Nicola Bertoldi. 2018. Escape: a large-scale synthetic corpus for automatic post-editing. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of association for machine translation in the Americas*, volume 200.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

POSTECH-ETRI's Submission to the WMT2020 APE Shared Task: Automatic Post-Editing with Cross-lingual Language Model

Jihyung Lee[1], **WonKee Lee**[1], **Jaehun Shin**[1]
Baikjin Jung[1], **Young-Kil Kim**[3], **Jong-Hyeok Lee**[1,2]
[1]Departmet of Computer Science and Engineering,
[2]Graduate School of Artificial Intelligence,
Pohang University of Science and Technology (POSTECH), Republic of Korea
[3]Electronics and Telecommunications Research Institute, Republic of Korea
{[1]jihyung.lee, [1]wklee, [1]jaehun.shin, [1]bjjung, [1,2]jhlee}@postech.ac.kr
[3]kimyk@etri.re.kr

Abstract

This paper describes POSTECH-ETRI's submission to WMT2020 for the shared task on automatic post-editing (APE) for 2 language pairs: English–German (En–De) and English–Chinese (En–Zh). We propose APE systems based on a cross-lingual language model, which jointly adopts translation language modeling (TLM) and masked language modeling (MLM) training objectives in the pre-training stage; the APE models then utilize jointly learned language representations between the source language and the target language. In addition, we created 19 million new sythetic triplets as additional training data for our final ensemble model. According to experimental results on the WMT2020 APE development data set, our models showed an improvement over the baseline by TER of -3.58 and a BLEU score of $+5.3$ for the En–De subtask; and TER of -5.29 and a BLEU score of $+7.32$ for the En–Zh subtask.

1 Introduction

Automatic post-editing (APE) is a subtask of MT, which aims to improve MT outputs by directly modifying machine-translated sentences (Chatterjee et al., 2019). Using APE systems to correct such errors that are automatically detectable can greatly reduce human effort compared to correcting machine-translated sentences manually from scratch (Pal et al., 2016).

Given that neural-network systems require a large quantity of training data, creating APE triplets, which each consist of a source sentence (`src`), a machine-translated sentence (`mt`), and a manually post-edited sentence (`pe`), requires a lot of human labor. Furthermore, because neural APE is a recently minted field of study, only a few small-sized training data sets are available at present. To mitigate such data shortage, several methods are proposed such that 1) create artificial APE triplets (Junczys-Dowmunt and Grundkiewicz, 2016; Negri et al., 2018); and 2) apply 'transfer learning' (Correia and Martins, 2019; Lopes et al., 2019). We believe that pre-trained models such as ELMo (Peters et al., 2018), OpenAI GPT (Radford et al., 2018), and BERT (Devlin et al., 2019) helped APE models learn rich language representations that compensated for the performance loss caused by using an insufficient quantity of training data.

APE is a task that handles both `src` and `mt` simultaneously, and learning a joint representation of these two inputs requires an understanding of both languages. Although previous works that used BERT have shown that transfer learning is effective in APE (Correia and Martins, 2019; Lopes et al., 2019), adopting BERT as a pre-trained language model may restrict to properly model the relation between two different languages because BERT is trained only on monolingual data sets. Therefore, following the recent trend of adopting transfer learning to various NLP tasks, we propose a new method that adopts a cross-lingual language model as a pre-trained langauge model for APE.

2 Related Work

2.1 APE models using BERT-based Encoder-Decoder

Lopes et al. (2019) proposed an APE system to which transfer learning is applied; the system uses multilingual BERT (Devlin et al., 2019) as its pre-trained language model in a Transformer encoder-decoder structure. They also introduced "conservativeness penalty", which discourages the APE system from frequently editing `mt`, into the system. In addition to using BERT as a cross-lingual encoder, they followed Correia and Martins (2019), which used pre-trained BERT to initialize weights of both the encoder and decoder, and shared weights of the

Proceedings of the 5th Conference on Machine Translation (WMT), pages 777–782

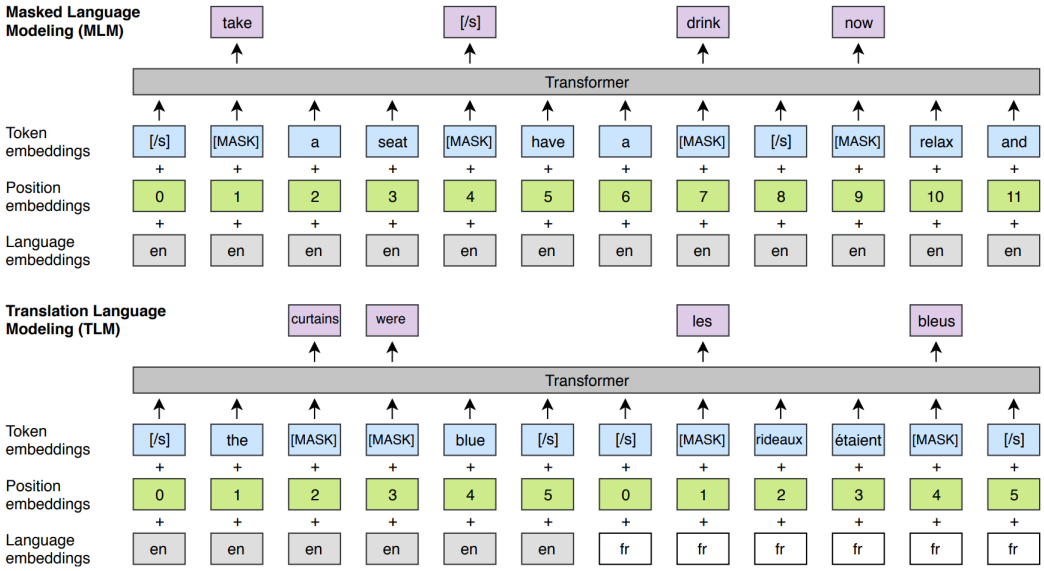


Figure 1: A comparison of the MLM objective and the TLM objective, taken from Conneau and Lample (2019)

self-attention layers both in the encoder and in the decoder.

Furthermore, they used a single encoder that accepts the concatenation of `src` and `mt` as input. To distinguish between the languages, they assigned different segment-embeddings for each language. This BERT-based encoder-decoder system showed the best performance for the English–German (En–De) language pair among all submissions for the WMT2019 APE shared task, proving the effectiveness of transfer learning.

2.2 XLM

After BERT had proposed masked language modeling (MLM), which requires monolingual data only (Devlin et al., 2019), Conneau and Lample (2019) introduced translation language modeling (TLM), which is an extension of MLM and allows the model to use parallel corpora as its input in the pre-training stage; the model can mask any token regardless of its language, and constructs its embedding by considering both sides of the context (Figure 1). The model learns through this process a cross-lingual representation during the training phase.

Considering that the APE task is a cross-lingual task, we expect that learning a cross-lingual representation of two different languages at the pre-training stage will be effective also in APE. Thus, we built a cross-lingual language model, which directly learns the joint representation of the two languages while being trained for the TLM objective, and we supplied it to our system. We describe our proposed model's architecture in the next section.

3 Model Description

Our APE system is built on top of Transformer's encoder-decoder structure (Vaswani et al., 2017). In the following subsections, we describe the main features of the encoder and decoder, respectively. Figure 2 illustrates the overall structure of our model.

3.1 Encoder

Transfer Learning. We built a cross-lingual language model and adopted this pre-trained language model to the encoder. It contains bidirectional and cross-lingual representations of the source and target languages, which are learned from predicting masked tokens from a big quantity of parallel data. Although a MLM+TLM model that was trained in 15 languages has been already released on the XLM GitHub page[1], to use a model that is trained with specific language pair corresponding to `src` and `mt` only, we built new MLM+TLM models. For En–De, we trained our model with the TLM objective on the pre-trained En–De MLM model which is released on the XLM Github page. For En–Zh, we trained our model with both the MLM and TLM objectives from scratch.

[1] https://github.com/facebookresearch/XLM

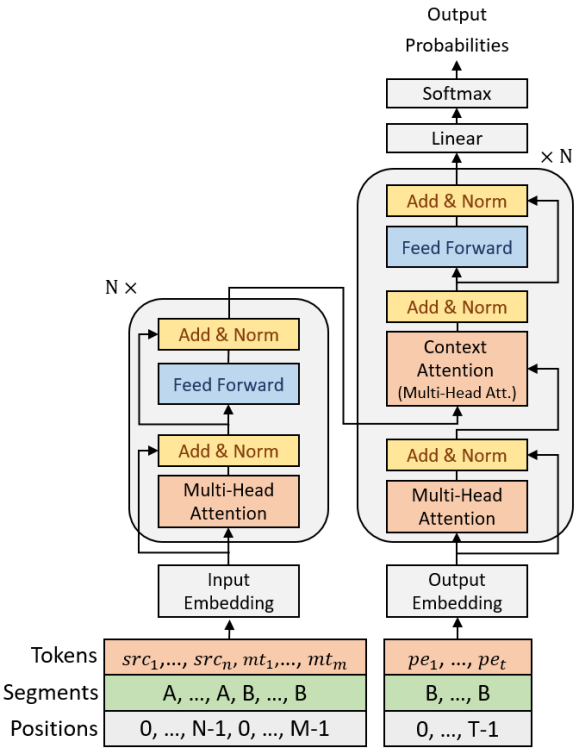


Figure 2: The architecture of our proposed APE model

Input representation. Unlike APE models that use a multi-source encoder that encodes `src` and `mt` separately (Junczys-Dowmunt and Grundkiewicz, 2018; Lee et al., 2019), we followed Lopes et al. (2019) so that the concatenation of `src` and `mt` was fed in to a single encoder. To distinguish one language from the other, we assigned different segment-embeddings to `src` and `mt`, respectively, and we also assigned individual positional-embeddings to `src` and `mt`.

3.2 Decoder

Because our pre-trained language model does not have a decoder, between two options, either randomly initializing the decoder or using another set of pre-trained weights, we chose the former; in contrast to Correia and Martins (2019), who made the encoder's self-attention weights be shared with the decoder, we randomly initialized the context attention layers and did not make the encoder and decoder share their parameters. To compensate for resulting variations in performance, we made an ensemble model of three to four individual models that have identical structures.

Reference	Corpus	En–De	En–Zh
WMT2020 News Translation Task	Europarl v10	✓	–
	ParaCrawl v5.1	✓	–
	Tilde RAPID	✓	–
	Tilde EESC	✓	–
	News Commentary v15	✓	✓
	WikiMatrix	✓	✓
	UN Parallel Corpus	–	✓
	Back-translated news	–	✓
OPUS	Wikipedia	✓	–
	MultiUN	✓	✓
	QED	✓	✓
WMT2019 QE Task Parallel Corpus		✓	–

Table 1: The list of data sets we used to train TLM in the pre-training stage for the En–De & En–Zh language pairs. All data sets were filtered to contain only such sentences with a length between 3 and 70 tokens.

4 Experiments

4.1 Dataset

We applied Byte-Pair Encoding (Sennrich et al., 2016) to all the corpora in both the source and target language. We used the En–De shared sub-word vocabulary that is released on XLM GitHub, but we compiled an En–Zh shared vocabulary by using Wikipedia's dump files in English and Chinese. As in the WMT2020 official data, all English and German data sets were truncated and tokenized with Moses (Koehn et al., 2007) scripts, and the Chinese data set was tokenized with the Jieba tokenizer.[2]

4.1.1 Pre-training stage

We collected parallel corpora from the WMT2020 News Translation Task website,[3] OPUS,[4] and the WMT2019 Quality Estimation website.[5] Table 1 shows the list of parallel corpora that we used to pre-train our models for the two language pairs. To build a pre-trained language model for En–Zh, we built a MLM+TLM model from scratch because we did not have available MLM models that are trained only on the English and Chinese data. Whereas we trained TLM on the whole parallel corpora, we trained MLM only using each side of the parallel corpora as monolingual data. For En–De, we trained only TLM; we used the En-De pre-trained MLM model that is released on the XLM Github page. The sizes of the final parallel corpora that we used in the pre-training stage are 51.7M triplets for En–De and 43.8M for En-Zh.

[2]https://github.com/fxsjy/jieba
[3]http://www.statmt.org/wmt20/translation-task.html
[4]http://opus.nlpl.eu/
[5]http://www.statmt.org/wmt19/qe-task.html

	English-German				English-Chinese			
	WMT20 Dev		WMT20 Test		WMT20 Dev		WMT20 Test	
	TER	BLEU	TER	BLEU	TER	BLEU	TER	BLEU
Baseline	31.36	50.37	31.56	50.21	60.41	22.62	59.49	23.12
Single	28.39	54.81	–	–	56.25	28.68	–	–
Primary - Top3Ens	**27.78**	**55.67**	27.37	55.83	**55.12**	**29.94**	**54.92**	28.90
Contrastive - Top4Ens	27.94	55.61	**27.02**	**56.37**	55.74	29.69	55.08	**28.97**

Table 2: TER and BLEU scores for En–De and En–Zh language pairs. APE results for the WMT2020 test data will be provided by the shared task organizers. 'Single' is the model which showed the best performance among all the models that later became constituents of the ensemble model.

4.1.2 APE training stage

We used the WMT2018 and WMT2020 official APE data sets for En–De, and the WMT2020 official APE data sets for En–Zh. As supplementary training data, we created new synthetic triplets by following the method to make the eSCAPE NMT data set (Negri et al., 2018); we used the parallel corpora that are released as additional resources for the WMT2020 Quality Estimation task.[6] To create those triplets, we first reused each side of the parallel corpora as `src` and `pe`. We then applied the QE NMT model (Fomicheva et al., 2020)[7] to `src` and then used the resulting translations as `mt`. As a result, we obtained 19M new synthetic triplets for both En–De and En–Zh.

4.2 Training Details

We modified Facebook's XLM implementation that is released on Github[8] to adapt it for the APE task. Most hyperparameters such as the number of layers, the hidden size, and the number of attention heads, were set to those that XLM used for the MT task (Conneau and Lample, 2019). We then used different optimizer-settings for the pre-training and APE training stage, respectively. We used the Adam optimizer (Kingma and Ba, 2014) with a learning rate of 5×10^{-5} in the pre-training stage, and 1×10^{-4}, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1 \times 10^{-6}$ in the APE training. We used 30k warm-up steps and a batch size of 32.

Similar to Lee et al. (2019), we divided the APE training process into two parts. The first is to train the model with 19M triplets, consisting of the 15-times up-sampled WMT official training data and our new synthetic data; we also added the WMT2018 official training data without up-sampling only for En-De. This first part took about three days on a single Tesla V100 GPU. The second is to fine-tune the model using only 7k triplets, which are official WMT2020 APE data. This second part took about three hours on the same GPU.

In the decoding stage, we used beam decoding with a beam size of five. We randomly initialized the weights of decoder's context attention layers and experimented our models four times to form an ensemble model of those four models. Our primary model is an ensemble model of three models, excluding one model that scored worst in terms of TER; this model showed the best performance on the WMT2020 development data set. Our contrastive model (Top4Ens) is an ensemble model of all the four models.

4.3 Results

We evaluated our results by comparing them to the MT baseline, which is uncorrected outputs of MT system. We used two evaluation methods that the WMT2020 APE task organizers suggested: Translation Error Rate (TER) and Bilingual Evaluation Understudy (BLEU). We used tercom software[9] to measure TER and a script of XLM GitHub to measure BLEU.

Table 2 describes the results of our proposed model on the WMT2020 official development and test data sets. For the development data set, our 'single' model outperformed the MT baseline in both language pairs. This result implies that our model successfully enhances the original quality of `mt`. Moreover, our primary ensemble model (Top3Ens) showed improvements over the MT baseline: for En–De by TER of -3.58 and by a BLEU score of

[6] http://www.statmt.org/wmt20/quality-estimation-task.html

[7] https://github.com/facebookresearch/mlqe/tree/master/nmt_models

[8] https://github.com/facebookresearch/XLM

[9] http://www.cs.umd.edu/ snover/tercom/

+5.3 and for En–Zh by TER of −5.29 and a BLEU score of +7.32.

Especially, for the test data set, our contrastive ensemble model showed a significant improvement for En–De by TER of −4.54 and a BLEU score of +6.16. For En–Zh, our primary submission showed an improvement over the MT baseline by a big margin: TER of −4.57 and a BLEU score of +5.78.

Although we submitted Top3Ens as our primary model, Top4Ens showed better TER and BLEU scores on the En–De test data set. We speculate that this result may have been caused by generality problem in which certain differences between the WMT2020 development and test data sets could occur.

5 Conclusion

For the WMT2020 APE shared task, we propose APE systems that adopt cross-lingual pre-trained language models. To better apply transfer learning to the APE task, we trained TLM in addition to using the original MLM models and initialized the decoder's weights in the same way as the encoder. Furthermore, we created new synthetic triplets to augment the training data and used the ensemble technique to build our final model.

Experimental results show that our proposed model achieved significant improvements on the WMT2020 development and test data sets in terms of TER and BLEU scores for both En–De and En–Zh language pairs.

Acknowledgments

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MISP) (No. 2019-0-01906, Artificial Intelligence Graduate School Program (POSTECH) and R7119-16-1001, Core technology development of the real-time simultaneous speech translation based on knowledge enhancement), and was results of a study on the "HPC Support" Project, supported by the 'Ministry of Science and ICT' and NIPA.

References

Rajen Chatterjee, Christian Federmann, Matteo Negri, and Marco Turchi. 2019. Findings of the WMT 2019 shared task on automatic post-editing. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 11–28, Florence, Italy. Association for Computational Linguistics.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems*, pages 7059–7069.

Gonçalo M. Correia and André F. T. Martins. 2019. A simple and effective approach to automatic post-editing with transfer learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3050–3056, Florence, Italy. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT (1)*.

Marina Fomicheva, Shuo Sun, Lisa Yankovskaya, Frédéric Blain, Francisco Guzmán, Mark Fishel, Nikolaos Aletras, Vishrav Chaudhary, and Lucia Specia. 2020. Unsupervised quality estimation for neural machine translation.

Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2016. Log-linear combinations of monolingual and bilingual neural machine translation models for automatic post-editing. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 751–758, Berlin, Germany. Association for Computational Linguistics.

Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2018. MS-UEdin submission to the WMT2018 APE shared task: Dual-source transformer for automatic post-editing. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 822–826, Belgium, Brussels. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

WonKee Lee, Jaehun Shin, and Jong-Hyeok Lee. 2019. Transformer-based automatic post-editing model with joint encoder and multi-source attention of decoder. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 112–117, Florence, Italy. Association for Computational Linguistics.

António V Lopes, M Amin Farajian, Gonçalo M Correia, Jonay Trénous, and André FT Martins. 2019. Unbabel's submission to the wmt2019 ape shared task: BERT-based encoder-decoder for automatic post-editing. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 118–123.

Matteo Negri, Marco Turchi, Rajen Chatterjee, and Nicola Bertoldi. 2018. eSCAPE: a large-scale synthetic corpus for automatic post-editing. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Santanu Pal, Sudip Kumar Naskar, and Josef van Genabith. 2016. Multi-engine and multi-alignment based automatic post-editing and its impact on translation productivity. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2559–2570, Osaka, Japan. The COLING 2016 Organizing Committee.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of NAACL-HLT*, pages 2227–2237.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Noising Scheme for Data Augmentation in Automatic Post-Editing

WonKee Lee[1], **Jaehun Shin**[1],
Baikjin Jung[1], **Jihyung Lee**[1], **Jong-Hyeok Lee**[1,2]
[1]Departmet of Computer Science and Engineering,
[2]Graduate School of Artificial Intelligence,
Pohang University of Science and Technology (POSTECH), Republic of Korea
{wklee, jaehun.shin, bjjung, jihyung.lee, jhlee}@postech.ac.kr

Abstract

This paper describes POSTECH's submission to WMT20 for the shared task on Automatic Post-Editing (APE). Our focus is on increasing the quantity of available APE data to overcome the shortage of human-crafted training data. In our experiment, we implemented a noising module that simulates four types of post-editing errors, and we introduced this module into a Transformer-based multi-source APE model. Our noising module implants errors into texts on the target side of parallel corpora during the training phase to make synthetic MT outputs, increasing the entire number of training samples. We also generated additional training data using the parallel corpora and NMT model that were released for the Quality Estimation task, and we used these data to train our APE model. Experimental results on the WMT20 English-German APE data set show improvements over the baseline in terms of both the TER and BLEU scores: our primary submission achieved an improvement of -3.15 TER and +4.01 BLEU, and our contrastive submission achieved an improvement of -3.34 TER and +4.30 BLEU.

1 Introduction

There has been a surge of interest in developing Automatic Post-Editing (APE) models, which is capable of automatically correcting errors produced by a machine-translation (MT) system, and thus is an attractive way to improve the quality of the MT output. Currently, sequence-to-sequence modeling has become a dominant approach to constructing APE models (Chatterjee et al., 2019, 2018), which requires a large quantity of training samples. However, APE data[1] — comprising triplets of three texts: source (*src*), a machine-translation (*mt*) of *src*, and a human-crafted post-edited sentence (*pe*) of *mt* — is too small and costly to acquire. Consequently, the lack of APE data becomes a great obstacle to a satisfactory performance of sequence-to-sequence models.

To reduce such data scarcity, there have been several attempts at constructing synthetic APE data (Negri et al., 2018; Junczys-Dowmunt and Grundkiewicz, 2016). Most notably, Negri et al. (2018) proposed a simple but effective way to construct a large-scale synthetic APE data set eSCAPE (*src*, *mt*, *ref*), of which *src* and *ref* is the source and target text of freely available parallel corpora, respectively, and *mt* is a translation of *src* produced by the MT system that had been trained on those parallel corpora.

As eSCAPE has shown to be beneficial in training APE models (Chatterjee et al., 2019), it has become feasible to train deep APE models and also what most recent works have been relying on so far. Nevertheless, the availability of a limited quantity of parallel corpora may not only be insufficient, but also vary depending on the language pair, that is, while some language pairs have plenty of resources, some others have relatively few resources. We thus argue that further works to supply additional resources should be needed to mitigate the potential data scarcity.

In this work, we introduce a noising scheme by which corrupted texts (ref_{noise}) are produced from *ref* of parallel corpora, resulting in additional APE triplets (src, ref_{noise}, ref), where *src* and *ref* is the source and target text of parallel corpora, respectively. During post-editing, certain editing operations including the word insertion, deletion, substitution, and shifting are applied to translated texts (noisy texts) for error correction. Thus, we applied such operations to target texts (clean texts) of parallel corpora to inject errors in reverse. Moreover, to simulate the quantity of errors that the target MT system produces, we refer to the distribution of "Translation Error Rate" (TER) (Snover et al., 2006) occurring in the actual APE data to determine the quantity of errors to be injected.

[1]http://www.statmt.org/wmt20/ape-task.html

Proceedings of the 5th Conference on Machine Translation (WMT), pages 783–788
Online, November 19–20, 2020.

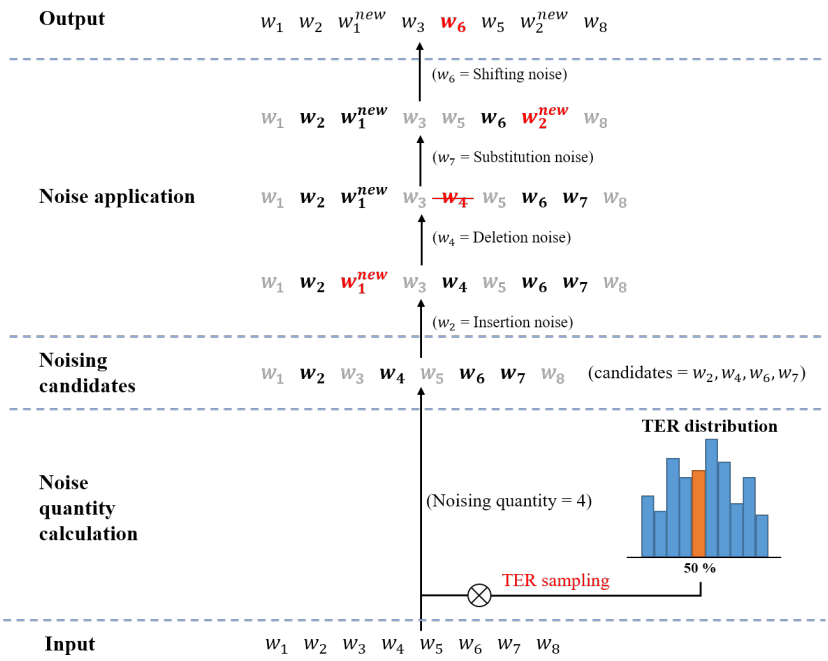


Figure 1: An illustration of the noising procedure

While we trained our models using the noising module, we supplied to the models synthetic APE data in addition to the WMT'20 APE data set as training data. The synthetic data was produced by using the eSCAPE method, which uses parallel corpora and a trained NMT model. We observed that models with noising module improved up to about -0.7 TER and +1.45 BLEU on the English-German (EN-DE) WMT'20 APE validation set compared to models without noising. Finally, our primary and contrastive submission to the WMT'20 APE shared task respectively recorded 28.41 TER and 54.22 BLEU, and 28.22 TER and 54.51 BLEU on the blind EN-DE WMT'20 APE test set.

2 Related Work

Noise injection to input sentences has become a popular method to let auto-encoders (Hill et al., 2016; Vincent et al., 2008) or pre-trained language models (Lewis et al., 2019; Devlin et al., 2019) learn how to reconstruct the original input. Because post-editing is a process of reconstructing corrupted translations, simulating corrupted MT outputs by injecting noise to the target sentence is a way to get synthetic APE training samples.

In the APE task, Xu et al. (2019) employed a data noising technique that incorporates a noise vector generated from a Gaussian or uniform distribution into the word embedding vector. However, their noising process has an effect on all tokens in a sequence, whereas only certain tokens in a given MT output are to be corrected in the APE process.

3 Method

3.1 Post-Editing Noise

Post-editing of *mt* texts requires four editing operations: insertion, deletion, substitution, and shifting. In other words, *mt* texts contain the following types of errors (The examples on the left side and right side are *mt* and *pe*, respectively.):

- Insertion operation implies that *mt* includes **deletion** errors:
  `We _ the world` → `We are the world`
- Deletion operation implies that *mt* includes **insertion** errors:
  `We are in the world` → `we are the world`
- Substitution operation implies that *mt* includes **substitution** errors:
  `We is the world` → `we are the world`
- Shifting operation implies that *mt* includes **shifting** errors:
  `We the world are` → `we are the world`

Considering the characteristics of editing operations, applying these operations to a clean text can simulate a corrupted *mt* text that can be post-edited to the original text. Thus, we corrupted a portion of words in a target text of parallel corpora, yielding a

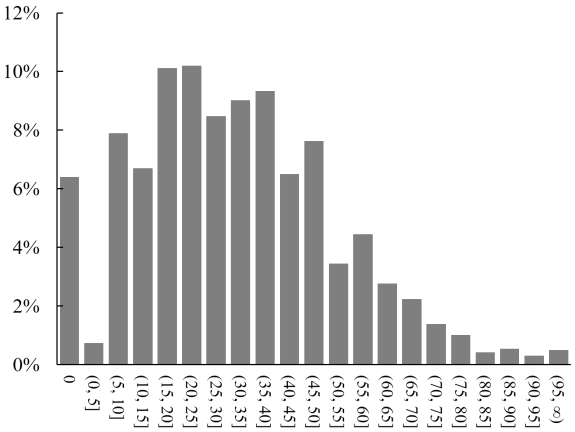


Figure 2: The categorical TER distribution of the WMT'20 training data, representing the proportion y [%] of samples belong to a specific TER range x.

new synthetic *mt* text which form a new synthetic triplet together with the corresponding source and target text in the parallel corpora.

3.2 Noising Procedure

Given an input sentence, we consider a noising procedure (Figure 1) (1) that specifies the quantity of words to become noise; (2) that selects specific words that will become noise according to the specified quantity; (3) and that determines the types of noise to be injected into the selected words.

Noising Quantity and Candidate Selection

The first step is to specify the quantity of words that will become noise in a given input sentence. Choosing a static proportion can be one option (Devlin et al., 2019), but imitating the proportion of errors that the target MT system produces would be more helpful to simulate the original *mt* text, considering that APE aims to correct the output produced by a particular MT system.

Accordingly, we refer to TER scores between *mt* and *pe* of the WMT'20 train set, which indicate the proportion of errors in *mt* that need to be corrected. Specifically, a TER range (e.g. $(45, 50]$) is drawn from the TER distribution in intervals of 5 (Figure 2), and then a specific value (e.g. 48) uniformly sampled from that range will be used as the error rate (e.g. $48 \rightarrow 0.48$). Finally, the noising quantity is calculated by multiplying the error rate by the input length. After specifying the noising quantity, we randomly select noising candidates among words in a given sentence according to the specified quantity.

Noise Application

Once the noising candidates have been selected, we now need to determine the types of noise that will be assigned to each candidate, and this process relies entirely on randomness. In particular, we produce four random numbers, making their summation equal to the noising quantity, and then this numbers are used as the quantity for each of the 'insertion', 'deletion', 'substitution', and 'shifting' post-editing noise. According to the quantity of each noising type, each type of noise is applied to words that are randomly selected among the noising candidates. Here, we present an example scenario as follows:

- Suppose that the noising quantity is 5 and the noising candidates are $w_1, w_4, w_7, w_9, w_{11}$ where w_i represents an input word in the i-th position.
- Given that the randomly selected numbers are $\{1, 2, 0, 2\}$, according to each of these four selected numbers, the words are randomly selected among the noising candidates, forming four subsets of selected words. (e.g.: $\{\{w_4\}, \{w_1, w_9\}, \{\emptyset\}, \{w_7, w_{11}\}\}$).
- Finally, one corresponding noise operation is applied to each subset of selected words. e.g.: $\{w_4\}$: insertion noise, $\{w_1, w_9\}$: deletion noise, $\{\emptyset\}$: substitution noise, $\{w_7, w_{11}\}$: shifting noise.

4 Experiment

4.1 Setup

Data. We collected publicly available parallel corpora that are listed on the WMT'20 Quality Estimation (QE) task webpage[2], which had been used to train the MT system by which *mt* texts of the WMT'20 QE corpus had been produced, and then we generated about 20M synthetic APE triplets (*src*, *mt*, *ref*) in the same manner as the eSCAPE method by using the pretrained MT system[3] that has been released on the QE task webpage. This synthetic APE triplets were used together with the official APE train set to train our APE model and a BPE model[4] by which we obtained a shared vocabulary of 32K subwords. During the training phase, the collected parallel corpora were used to

[2]http://www.statmt.org/wmt20/quality-estimation-task.html

[3]https://github.com/facebookresearch/mlqe

[4]https://github.com/google/sentencepiece

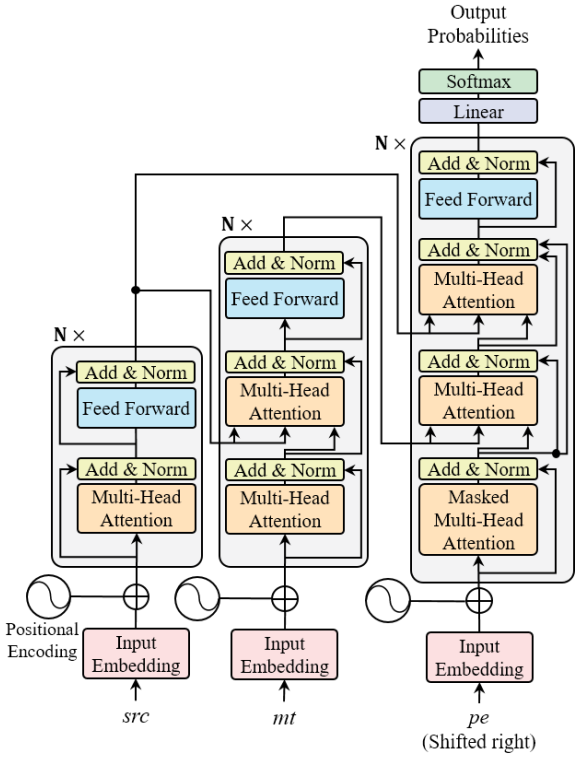


Figure 3: The structure of our APE model.

generate another synthetic APE triplets produced by our noising approach.

Model configuration. We adopted the "sequential APE model" proposed by Lee et al. (2019) to construct our APE model (Figure 3) by applying some minor modifications: the ReLU activation (Nair and Hinton, 2010) in the 'feed-forward' layers was replaced with the GELU activation (Hendrycks and Gimpel, 2016), an additional residual connection between the outputs of the first and third multi-head attention layer was added in the decoder. Additionally, we removed some details such as "stack-level attention" and "future masking to *mt*". We set our model's hyperparameters as follows: the size of the word embedding and all hidden dimensions at 768; the size of the inner dimension of feed-forward layers at 3072; 8 heads; 6 layers; and dropout rate at 0.1. The model was optimized using Adam (Kingma and Ba, 2015) with $\beta_1 = 0.9, \beta_2 = 0.998$, and $\epsilon = 1e^{-8}$, and we used the same learning rate scheduling as Vaswani et al. (2017) with 15,000 warmup steps. We implemented and trained our model by using the OpenNMT-py[5] framework.

[5]https://github.com/OpenNMT/OpenNMT-py

Model	WMT'20 dev		WMT'20 test	
	TER	BLEU	TER	BLEU
Baseline	31.37	50.37	31.56	50.21
APE_{base}	29.42	52.32	–	–
+Noise	28.72	53.77	–	–
Submission (Ensemble)				
Primary	28.70	53.77	28.41	54.22
Contrastive	–	–	28.22	54.51

Table 1: The evaluation results. APE_{base} stands for the model trained without using our noising approach.

4.2 Training Details

Training procedure. Training of the APE model is composed of two steps. At the first step, both the synthetic triplets and WMT'20 training data are used to train the model. Every time each training batch is assigned to the model, 25% of the synthetic triplets (*src*, *mt*, *ref*) in the batch are replaced with another synthetic triplets (*src*, ref_{noise}, *ref*) by applying the noising procedure (§3.2) to each *ref*. We here set the batch size at 33,000 tokens. The following step is to fine-tune the model by only using the WMT'20 data, starting at the convergence point found in the first step. At this step, we set the batch size at 1,024 tokens.

Ensemble. We trained two ensemble models for submission. Our primary submission (TERNoise-Ops-Ens8) is an ensemble of eight runs. We first selected the top five runs, which had the lowest TER on the development set, for three individual weight initializations. To form the ensemble model, we then selected among them the top two runs, for each of four edit operations, that make corrections most frequently. Our contrastive submission (TERNoise-nFold-Ens8) is also an ensemble of eight runs. Aiming for generalization to unseen data, all runs were trained and validated in a 4-fold setting on a data set into which the training data and development data had been merged. Then we selected the top two runs for each fold to form the ensemble model.

4.3 Result

Table 1 presents our evaluation results. As the WMT20 test data is blind to users, we were not able to conduct an evaluation on the test data set, but we observed that applying our noising scheme improved the post-editing quality of the model on the development data set. As a result, our primary submission, which is an ensemble of models adopt-

ing the noising scheme, showed an improvement of -3.15 TER score and $+4.01$ BLEU score on the test data set. In addition, our contrastive submission, which is an ensemble of models trained on the separate training data set to seek generality performance on unseen data, showed better results than our primary submission, resulting in its high generalization capability to the unseen the WMT20 test data.

5 Conclusion

We propose a noising scheme to supply APE models with synthetic APE triplets during the training phase. Our noising scheme is designed based on the error types that are defined in the APE task, and the quantity of noise that are injected during the training phase are determined in consideration of the distribution of those error types in the official training data. According to the experimental results, applying our noising scheme to APE models showed an improvement of the post-editing quality in terms of both TER and the BLEU scores, which indicates that our noising scheme was effective in training APE models although there may be differences between the synthesized errors and the actual MT errors. Therefore, in future work, we will aim to reduce those gaps caused by our noising scheme.

Acknowledgments

This work was supported by Institute for Information & communications Tecnology Promotion (IITP) grant funded by the Korea government (MISP) (No. 2019-0-01906, Artificial Intelligence Graduate School Program (POSTECH) and R7119-16-1001, Core technology development of the real-time simultaneous speech translation basd on knowledge enhancement), and was results of a study on the "HPC Support" Project, supported by the 'Ministry of Science and ICT" and NIPA.

References

Rajen Chatterjee, Christian Federmann, Matteo Negri, and Marco Turchi. 2019. Findings of the WMT 2019 shared task on automatic post-editing. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 11–28, Florence, Italy. Association for Computational Linguistics.

Rajen Chatterjee, Matteo Negri, Raphael Rubino, and Marco Turchi. 2018. Findings of the WMT 2018 shared task on automatic post-editing. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 710–725, Belgium, Brussels. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Dan Hendrycks and Kevin Gimpel. 2016. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*.

Felix Hill, Kyunghyun Cho, and Anna Korhonen. 2016. Learning distributed representations of sentences from unlabelled data. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1367–1377.

Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2016. Log-linear combinations of monolingual and bilingual neural machine translation models for automatic post-editing. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 751–758, Berlin, Germany. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

WonKee Lee, Jaehun Shin, and Jong-Hyeok Lee. 2019. Transformer-based automatic post-editing model with joint encoder and multi-source attention of decoder. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 112–117.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Vinod Nair and Geoffrey E Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *ICML*.

Matteo Negri, Marco Turchi, Rajen Chatterjee, and Nicola Bertoldi. 2018. Escape: a large-scale synthetic corpus for automatic post-editing. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of association for machine translation in the Americas*, volume 200. Cambridge, MA.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. 2008. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103.

Hongfei Xu, Qiuhui Liu, and Josef van Genabith. 2019. Uds submission for the wmt 19 automatic post-editing task. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 147–152, Florence, Italy. Association for Computational Linguistics.

Alibaba's Submission for the WMT 2020 APE Shared Task: Improving Automatic Post-Editing with Pre-trained Conditional Cross-Lingual BERT

Jiayi Wang*, **Ke Wang***, **Kai Fan, Yuqi Zhang, Jun Lu, Xin Ge, Yangbin Shi, Yu Zhao**
Alibaba Group Inc., Hangzhou, China
{joanne.wjy,moyu.wk,k.fan,chenwei.zyq,joelu.luj,
shiyi.gx,taiwu.syb}@alibaba-inc.com, kongyu@taobao.com

Abstract

The goal of Automatic Post-Editing (APE) is basically to examine the automatic methods for correcting translation errors generated by an unknown machine translation (MT) system. This paper describes Alibaba's submissions to the WMT 2020 APE Shared Task for the English-German language pair. We design a two-stage training pipeline. First, a BERT-like cross-lingual language model is pre-trained by randomly masking target sentences alone. Then, an additional neural decoder on the top of the pre-trained model is jointly fine-tuned for the APE task. We also apply an imitation learning strategy to augment a reasonable amount of pseudo APE training data, potentially preventing the model to overfit on the limited real training data and boosting the performance on held-out data. To verify our proposed model and data augmentation, we examine our approach with the well-known benchmarking English-German dataset from the WMT 2017 APE task. The experiment results demonstrate that our system significantly outperforms all other baselines and achieves the state-of-the-art performance. The final results on the WMT 2020 test dataset show that our submission can achieve +5.56 BLEU and -4.57 TER with respect to the official MT baseline.

1 Introduction and Related Work

Even machines can approach and achieve parity with human translations (Hassan et al., 2018) empowered by a sequence-to-sequence fashion (Bahdanau et al., 2014; Vaswani et al., 2017), post-editing is still an important and necessary step in the translation process, especially in scenarios where extremely high-quality translation results are essentially required such as business legal documents, technical product guides, medicine instructions and so on. It is the process whereby humans amend machine-generated translation to achieve an acceptable final product. Translation crowdsourcing paradigm, computer assisted translation (CAT) thus comes into being as demanded, which includes a hybrid of machine translation and human post-editing to meet translation scenarios with different quality requirements accordingly for accuracy, clarity, fluency, and domain adaptation.

However, post-editing, while improving, that can match human understanding of meaning, nuance, tone, humor–the list goes on, it's often worth paying extra more. The time spent on translation mistake corrections by humans remains substantial to the extent (Läubli et al., 2013) so that it even occasionally offsets the efficiency gained from the neural machine translation (NMT) systems. In this paper, we explore automatic post-editing (APE) in a deep learning framework where a two-stage training pipeline is engaged. The goal of APE task is to examine automatic methods of correcting translation mistakes produced by a black-box machine translation engine to improve the MT results. Human efforts are correspondingly reduced in the later editing process (Läubli et al., 2013) if our APE system can approach human translations as much as possible.

Traditionally, APE is a supervised learning task, requiring sufficient training data in the triplet of source (SRC), machine translation (MT) and post editing (PE) that are usually expensively available. Due to the limited number of such APE data released officially in this year's APE tasks and the specific domain, Wikipedia, which is quite different from the previous years'(IT domain), we adopt an imitation learning to mine WMT corpora, eSCAPE (Negri et al., 2018), Opus Wikipedia corpus (Wołk and Marasek, 2014) and our own English-German corpus to augment APE training data. However, pseudo data strategy is far from enough to train the state-of-the-art APE system. Inspired by the

* indicates equal contribution.

Proceedings of the 5th Conference on Machine Translation (WMT), pages 789–796
Online, November 19–20, 2020.

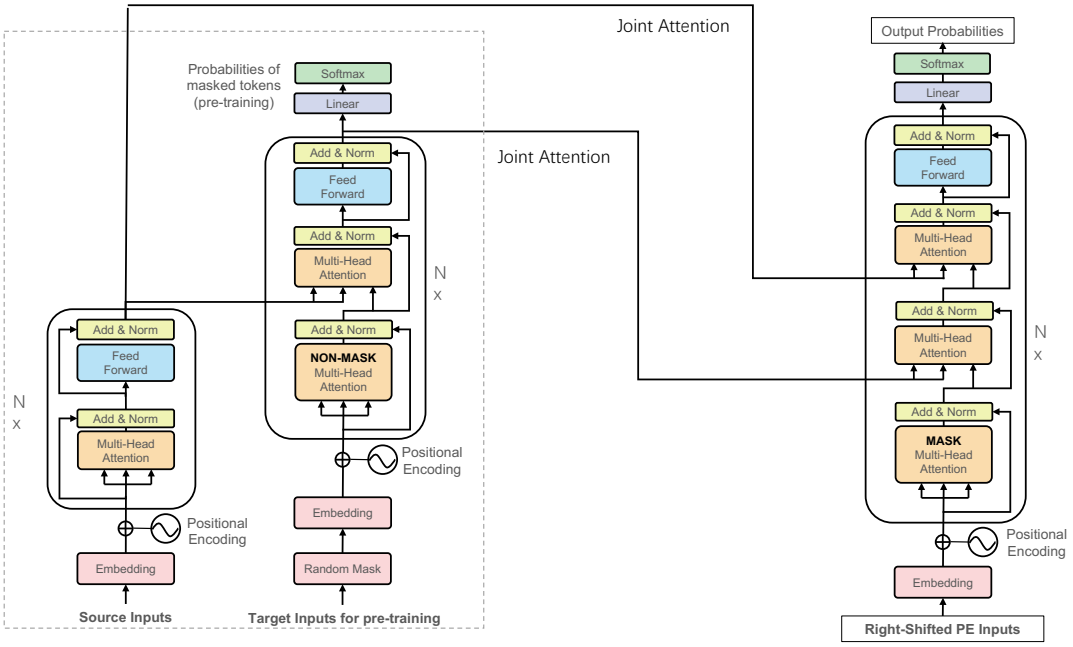


Figure 1: The APE model structure including detailed operations in pre-training and training

masked language model objective in the encoder BERT (Devlin et al., 2018), we introduce our Bert-like cross-lingual training objective to the encoder-decoder framework by adapting the decoder to become a memory encoder (Fan et al., 2019), allowing us to pre-train the target language model similar to BERT but conditioned on the source language text. Knowledge learned from the pre-training can be extensively transferred to many second-step downstream tasks, including but not limited to translation quality estimation, parallel corpus filtering and of course, automatic post-editing. The overall framework of our APE model is the same with the generative automatic post-editing model's structure in Wang et al. (2020).

Similar training mechanism is applied in the winner system of WMT 2019 APE Shared Task (Lopes et al., 2019), that wisely takes full advantage of the pre-trained multilingual BERT (mBERT) (Devlin et al., 2018) and achieves top performances.They concatenate the source and machine translation sentences to feed into the encoder mBERT and then fine tune the encoder and a transformer decoder where the context attention block is initialized by the self-attention weights of mBERT as well.

We examine our approach on the public English–German dataset from WMT 2017 APE shared task. Our system outperforms the top ranked methods in both BLEU (Papineni et al., 2002) and TER (Snover et al., 2006) metrics. For this year's WMT APE task, we finally submitted two ensemble English-German APE models according to different model selection methods and accomplish +5.56 increase in BLEU and -4.57 decline in TER on 2020 test set.

2 Methodology

In this section, we will introduce our APE model in terms of general structure and some computation details together with our data augmentation strategy.

2.1 APE Model Structure

The structure of our APE pre-training model originates from adapting the decoder in the transformer (Vaswani et al., 2017) to a memory encoder, following the exactly same design in Fan et al. (2019). We randomly pick 15% tokens in the target sentences during each training step to be substituted with a special [mask] token where predictions will be requisites accordingly, covering 12% masked, 1.5% substituted and 1.5% unchanged. In order to train a masked language model on the target sentences conditional on the sources, we accordingly remove the future mask matrix in the self-attention of the decoder to form a memory encoder, aiming to learn deep syntactic and alignment information of the ground truth. Therefore, during pre-training stage, the model is trained with high-quality English-German parallel corpora.

After we fully train the conditional language model on the target side, we apply an auto-regressive decoder on top of the pre-trained encoder-memory encoder model to decode the post-editing results in the stage of APE training by using the triplet data.

Figure 1 shows the details of our model structure. The part in the dotted line represents the pre-training stage with the removal of future mask in the memory encoder, and the whole picture describes the APE training process when the encoder-memory encoder pre-training model has been trained thoroughly. Note that the order of joint attentions of encoder and memory encoder with the decoder separately can be switched. Our experimental results in the following section illustrate this slight change can bring benefits to the diversity of the models and enhance the final ensemble's performance.

2.2 Data Strategies

High-quality parallel corpus filtering Our pre-training model requests high-quality parallel corpora. The dual conditional cross-entropy model (Junczys-Dowmunt, 2018) has been proven effective in WMT 2018 Corpus Filtering Shared Task. The cross-entropy scores according to two inverse translation models trained on clean data are used as the quality indicator so that we are able to mine qualified parallel sentences from noisy parallel corpora.

APE training data augmentation. Domain Adaption methods have been also investigated because of the small amount of official English-German APE training set and the special domain, Wikipedia. A semi-supervised CNN domain classification model (Chen and Huang, 2016) trained with in-domain seed and other general-domain data is utilized to extract in-domain source and target sentences from English-German corpora to augment pseudo sources and post-edits for APE training. To generate the corresponding machine translations of the classified in-domain source sentences, we use the rest of our corpus to train a neural machine translation model with model setting in Vaswani et al. (2017) to produce the MT results. The pseudo sources and post-edits are used as supplementary data during pre-training, and the pseudo triplets improve APE performance on the basis of only using official APE training set.

Algorithm 1 Imitation Learning for Fine-tuning

Require: Reference Set $\mathbf{R} = \{(s_i, m_i, e_i)\}_{i=1}^{M}$, Full Training Set $\mathbf{T} = \{(s_j, m_j, e_j)\}_{j=1}^{N}$, hyperparameters $K \in [1, +\infty)$, $\alpha \in (0, 1)$.

Set the output dataset $\mathbf{R} = \{\}$.
for each $(\mathbf{s_i}, \mathbf{m_i}, \mathbf{e_i})$ in R **do**
  $\vec{V_r} = (TER(e_i, m_i), Length(e_i))$
  Candidate Set $C = \{\}$
  for each $(\mathbf{s_j}, \mathbf{m_j}, \mathbf{e_j})$ in T **do**
    $\vec{V_t} = (TER(e_j, m_j), Length(e_j))$
    for m in 0,1 **do**
      if $\left\| (\vec{V_r}[m] - \vec{V_t}[m])/\vec{V_r}[m] \right\| > \alpha$ **then**
        Skip this training sample
      end if
    end for
    Add this training sample $(\mathbf{s_j}, \mathbf{m_j}, \mathbf{e_j})$ to C
  end for
  if size of $C > K$ **then**
    Sort candidates in C by its cosine similarity to $\vec{V_r}$
    Remain only the top K candidates in C
  end if
  for each candidate in C **do**
    Add it to F
    Remove it from T
  end for
end for
return Filtered Dataset F.

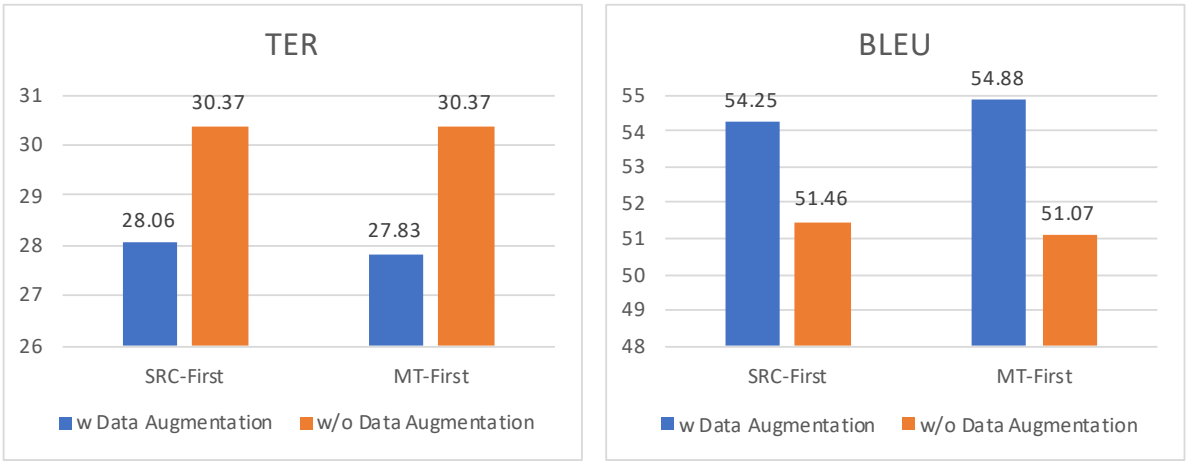
Imitation learning. To boost the APE model performance, we optimize our model during the APE training stage with further filtered APE data by an imitation learning method, since we noticed that there are gaps between the distributions of TERs in different types of our APE training set. Deeply motivated by Junczys-Dowmunt and Grundkiewicz (2016), we leverage the official training data containing real 7000 in-domain APE triplets as a reference set and apply Algorithm 1 to sample a subset of the whole training data in Table 1. Then we fine tune the APE model further with such a subset that has a similar distribution with this year's official training data. All the details of data usage will be described in the following experiment section.

3 Experiment

We conduct our experiments on two different datasets: First, to make a fair comparison with other top-ranked systems on WMT APE tasks in recent years, we perform a single model evaluation on the WMT 2017 English-German APE Shared Task without any other pseudo data except the Artificial dataset (Junczys-Dowmunt and Grundkiewicz, 2016) provided officially (for fair comparisons, and we avoid using the Escape Corpus (Negri et al., 2018) which has not been released until 2018); Second, we carry out a series of experiments

Real/Pseudo	MT Engine	In/Out Domain	Up-sample Weight	Description	Size
Real	SMT	Out-domain	10	Train set of WMT 16&17 APE task	23k
Real	NMT	Out-domain	20	Train set of WMT 18 APE task	13.4k
Real	NMT	In-domain	40	Train set of WMT 20 APE task	7k
Pseudo	SMT	Out-domain	1	Artificial Dataset	4.4M
Pseudo	NMT	Out-domain	1	Escape Corpus (NMT)	4.9M
Pseudo	NMT	In-domain	1	Our in-domain pseudo data	20M
Total	-	-	-	Final training set	30M

Table 1: Compositions of the Training Data for the WMT 2020 APE Shared Task



(a) TER with/without Data Augmentation

(b) BLEU with/without Data Augmentation

Figure 2: Results in the English-German Development Set of WMT 2020 APE Shared Task of Different Model Structures with/without Data Augmentation

on the WMT 2020 English-German APE Shared Task with strategies including data argumentation, quality filtering, domain adaptation, and model ensemble to accomplish the overall performance of our model.

3.1 Setup

Dataset. For the experiments on WMT 2017 APE, we verify our APE model design on the open public WMT 2017 English-German APE Shared Task (Ondrej et al., 2017). The official training set consists of 23K real triplets (SRC, MT, PE) for training and another 2K triplets for testing from the Internet Technology (IT) domain. Besides, the shared task offers a large-scale artificial synthetic corpus containing around 500K high-quality and 4 million relatively low-quality synthetic triplets. We over sample the APE real data by 20 times and merge it with the synthetic data, resulting in roughly 5 million of triplets for both pre-training and APE training. The final APE system is selected based on WMT 2016 APE test set.

For the experiments on WMT 2020 APE, we use all available APE triplets of WMT English-German APE tasks released since 2016, including about 43.4K real triplets as well as 9.3M synthesized data made up with Artificial (Junczys-Dowmunt and Grundkiewicz, 2016) and Escape (Negri et al., 2018). Considering the application domain for this year's task changes from IT to Wikipedia and the size of the official in-domain training set is quite small (only 7000 samples), we generate about 20M in-domain pseudo data for our model training as follows:

1. We apply the cross-entropy scoring algorithm described in section 2.2 on our own English-German parallel corpus and filter out about 200 million high-quality parallel data with a proper threshold.

2. We collect the Wikipedia corpus from Wołk and Marasek (2014), which contains more than 2 million of English-German parallel sentences. We up-sample the SRC of this year's training data 20 times and mix them with the English side of Wikipedia corpus as our in-

domain seeds and train a domain classification model as described in section 2.2 with other general-domain data including the news and biomedical dataset from the WMT 2020 website. Afterwards, the domain classfication model is applied to extract about 20 million of in-domain parallel sentences from the 200M high-quality parallel data mentioned above.

3. The left 180 million are used to train a English-German transformer-based neural machine translation model (Vaswani et al., 2017) with the OpenNMT (Klein et al., 2017) source code. The sources and targets of the 20M high-quality in-domain parallel corpus are treated as SRCs and PEs and the decoding results from the trained NMT model are regarded as corresponding MTs. These in-domain pseudo triplets are mixed with all available training set from the WMT APE Shared Task since 2016 with differentiated up-sample weights as our final training set, as shown in Table 1.

Pre-processing. In all of our experiments, we apply truecasers trained independently for English and German seperately (Koehn et al., 2007) and process our data into subword units (Kudo, 2018) with a 32K shared vocabulary. Triplets with more than 70 subword units in any one of the SRCs, MTs or PEs are removed.

Evaluation Metrics. We mainly evaluate our systems with the metrics, translation edit rate (TER) (Snover et al., 2006) and bilingual evaluation understudy (BLEU) (Papineni et al., 2002), since they are standard and widely employed in evaluation of the WMT APE tasks.

Model Setting. All experiments are trained on 8 NVIDIA P100 GPUs for maximum 100,000 steps for about two days until convergence, with a total batch-size of 65536 tokens per step and the Adam optimizer (Kingma and Ba, 2014). Parameters are being tuned with 12,000 steps of learning rates warm-up (Vaswani et al., 2017). Except these modifications, we follow the default transformer-based configuration (Vaswani et al., 2017) for other hyper-parameters settings.

3.2 Results on WMT 2017 APE Shared Task

We verify the validity and efficiency of our proposed model on WMT 2017 APE test data since all of the winners of WMT APE Shared Tasks of recent years do report their results of single models on this dataset (Junczys-Dowmunt and Grundkiewicz, 2018; Correia and Martins, 2019). To make a fair comparison, we do not use any extra data for training as described in the data setup.

The main results of APE systems are presented in Table 2, demonstrating that our single model, even without pre-training, outperforms all winners of the WMT APE Shared Task from 2017 to 2019 on both BLEU and TER metrics.

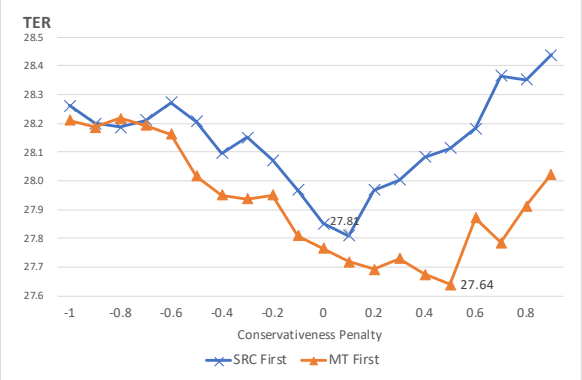


Figure 3: TERs on the English-German Development Set of WMT 2020 APE Shared Task from the Further Optimized Models with Different Values of Conservativeness Penalty

3.3 Results on WMT 2020 APE Shared Task

For this year's task, we adopt various of strategies including data augmentation, further optimization by imitation learning and model ensemble.

Data Augmentation As described in section 2.2, we utilize several algorithms, quality filtering and domain adaption, to construct our own in-domain pseudo data for APE training. We conduct experiments with and without in-domain pseudo data on two different model structures described in Section 2.1 for decoder joint attention switching (referred as SRC-First and MT-First respectively in the following discussion). Results on the 2020 development set in Figure 2 indicate that our data augmentation strategies can generate powerful pseudo data which significantly improve the model performance in this year's APE task.

Further Optimization via Imitation Learning The hyper-parameters α and K in Algorithm 1 are set to 0.3 and 500 according to empirical studies. Finally, around 2M triplets are filtered from the full training set via the imitation learning algorithm. We compare TERs before and after APE fine tun-

Model	BLEU↑	TER↓	Note
Official Baseline	62.49	24.48	Do nothing to the original machine translation
FBK (Ensemble)	70.07	19.60	Ensemble model, winner of WMT17 APE task
MS-UEdin	69.72	19.49	Single model, winner of WMT18 APE task
Unbabel (BED)	70.66	19.03	Single model, winner of WMT19 APE task.
Proposed Model w/o pre-training	70.90	18.90	Single model without pre-training
Proposed Model w pre-training	71.52	18.44	Single model with pre-training

Table 2: Performance Comparisons on WMT 2017 APE English-German Test Set

Model	BLEU↑	TER↓	Note
Official Baseline	50.37	31.37	Do nothing with the original machine translation
Ensemble×5 of BED	55.09	27.85	The winning system of last year
Our Single Model	54.88	27.83	MT-First structure
+ Optimizing	54.50	27.76	Optimized on filtered subset
+ Conservativeness Penalty	54.87	27.64	Conservativeness penalty = 0.5
Our Ensemble×5	55.87	27.02	Our contrastive submission
Our Ensemble×5	**56.06**	**26.99**	Our primary submission

Table 3: Main Results in the English-German Development Set of the WMT 2020 APE Shared Task

Model	BLEU↑	TER↓
Official Baseline	50.21	31.56
Our Primary Submission	55.58	27.03
Our Contrastive Submission	**55.77**	**26.99**

Table 4: Submission Results in the English-German Test Set of the WMT 2020 APE Shared Task

ing with the filtered data in Figure 4 with the two different model structures. It can be clearly shown that the APE model can be further improved.

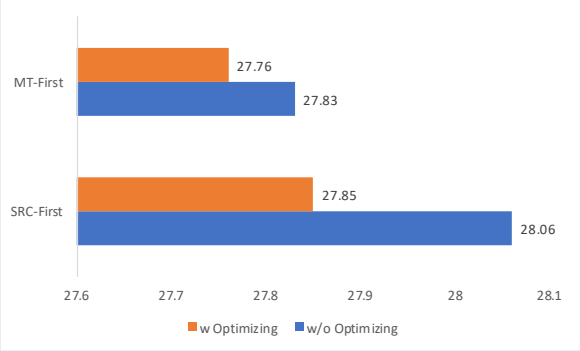


Figure 4: TERs on the English-German Development Set of WMT 2020 APE Shared Task for Different Model Structures with/without Further Optimizing

Ensemble We train the two different APE models (SRC-First & MT-First), each for three times with 30M APE training set to get 6 primary APE models and fine tune all of them with 2M filtered APE data via imitation learning for further optimization. Then, we obtain 12 APE models, 6 primary models and 6 optimized ones. Our final primary submission is an ensemble of the top 5 primary models with lowest TERs. In contrast, an ensemble of the top 5 optimized models is submitted as well for validation of imitation learning method.

Following the winning system of last year, we apply the conservativeness penalty (Lopes et al., 2019) on each model before ensemble. As shown in Figure 3, the local optimal solutions for the conservativeness penalty may be various among models. Therefore, instead of a fixed constant, we apply the most appropriate penalties for each model according to their performance on the 2020 development set. Results of our ensemble models in the development set and the test set can be found at Table 3 and Table 4 respectively.

Besides, we also train last year's winning system five times (BED (Lopes et al., 2019)) with the exactly same data we use for WMT 2020 APE task based on the source code they released[1] and pro-

[1]https://github.com/deep-spin/OpenNMT-APE

duce an ensemble result reported in Table 3. Evaluated on 2020 development set, both of our final ensemble model in primary and contrastive submissions outperform the winning system of 2019. The final results on the 2020 test set released officially show that our ensemble models significantly improve the machine translations with significant margins in TER and BLEU (-4.57 TER and +5.56 BLEU).

4 Conclusion

This paper describes our automatic post-editing system for the WMT 2020 English-German APE Shared Task. We introduce a cross-lingual Bert-like conditional model with an innovative memory encoder which can capture the deep semantic information of machine translations conditional on the source sentences. In addition, efforts on data augmentation strategies, corpus filtering and imitation learning, are able to overcome the scarcity of real APE data and further improve the model performance together with the ensemble strategy. Our single APE model outperforms all winner systems of recent years' WMT APE Shared Tasks on the WMT 2017 English-German test set and achieves impressive performances on the WMT 2020 English-German APE test set.

Acknowledgments

This work is partly supported by National Key RD Program of China (2018YFB1403202).

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Boxing Chen and Fei Huang. 2016. Semi-supervised convolutional networks for translation adaptation with tiny amount of in-domain data. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 314–323.

Gonçalo M. Correia and André F. T. Martins. 2019. A simple and effective approach to automatic post-editing with transfer learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3050–3056, Florence, Italy. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Kai Fan, Jiayi Wang, Bo Li, Boxing Chen, and Niyu Ge. 2019. Neural zero-inflated quality estimation model for automatic speech recognition system. *arXiv preprint arXiv:1910.01289*.

Hany Hassan, Anthony Aue, Chang Chen, Vishal Chowdhary, Jonathan Clark, Christian Federmann, Xuedong Huang, Marcin Junczys-Dowmunt, William Lewis, Mu Li, et al. 2018. Achieving human parity on automatic chinese to english news translation. *arXiv preprint arXiv:1803.05567*.

Marcin Junczys-Dowmunt. 2018. Dual conditional cross-entropy filtering of noisy parallel corpora. *arXiv preprint arXiv:1809.00197*.

Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2016. Log-linear combinations of monolingual and bilingual neural machine translation models for automatic post-editing. *arXiv preprint arXiv:1605.04800*.

Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2018. Ms-uedin submission to the wmt2018 ape shared task: Dual-source transformer for automatic post-editing. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 835–839. Association for Computational Linguistics.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M Rush. 2017. Opennmt: Open-source toolkit for neural machine translation. *arXiv preprint arXiv:1701.02810*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*, pages 177–180. Association for Computational Linguistics.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. *arXiv preprint arXiv:1804.10959*.

Samuel Läubli, Mark Fishel, Gary Massey, Maureen Ehrensberger-Dow, Martin Volk, Sharon O'Brien, Michel Simard, and Lucia Specia. 2013. Assessing post-editing efficiency in a realistic translation environment.

António V Lopes, M Amin Farajian, Gonçalo M Correia, Jonay Trenous, and André FT Martins. 2019. Unbabel's submission to the wmt2019 ape shared task: Bert-based encoder-decoder for automatic post-editing. *arXiv preprint arXiv:1905.13068*.

Matteo Negri, Marco Turchi, Rajen Chatterjee, and Nicola Bertoldi. 2018. Escape: a large-scale synthetic corpus for automatic post-editing. *arXiv preprint arXiv:1803.07274*.

Bojar Ondrej, Rajen Chatterjee, Federmann Christian, Graham Yvette, Haddow Barry, Huck Matthias, Koehn Philipp, Liu Qun, Logacheva Varvara, Monz Christof, et al. 2017. Findings of the 2017 conference on machine translation (wmt17). In *Second Conference onMachine Translation*, pages 169–214. The Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of association for machine translation in the Americas*, volume 200. Cambridge, MA.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Ke Wang, Jiayi Wang, Niyu Ge, Yangbing Shi, Yu Zhao, and Kai Fan. 2020. Computer assisted translation with neural quality estimation and automatic post-editing. *arXiv preprint arXiv:2009.09126*.

Krzysztof Wołk and Krzysztof Marasek. 2014. Building subject-aligned comparable corpora and mining it for truly parallel sentence pairs. *Procedia Technology*, 18:126–132.

HW-TSC's Participation at WMT 2020 Automatic Post Editing Shared Task

Hao Yang[1], **Minghan Wang**[1], **Daimeng Wei**[1], **Hengchao Shang**[1], **Jiaxin Guo**[1], **Zongyao Li**[1], **Lizhi Lei**[1], **Ying Qin**[1], **Shimin Tao**[1], **Shiliang Sun**[2], **Yimeng Chen**[1]
[1]Huawei Translation Services Center, Beijing, China
[2]East China Normal University, Shanghai, China
{yanghao30,wangminghan,weidaimeng,shanghengchao,guojiaxin1,
lizongyao,leilizhi, qinying,taoshimin,chenyimeng}@huawei.com
slsun@cs.ecnu.edu.cn

Abstract

The paper presents the submission by HW-TSC in the WMT 2020 Automatic Post Editing Shared Task. We participate in the English→German and English→Chinese language pairs. Our system is built based on the Transformer pre-trained on WMT 2019 and WMT 2020 News Translation corpora, and fine-tuned on the APE corpus. Bottleneck Adapter Layers are integrated into the model to prevent over-fitting. We further collect external translations as the augmented MT candidates to improve the performance. The experiment demonstrates that pre-trained NMT models are effective when fine-tuning with the APE corpus of a limited size, and the performance can be further improved with external MT augmentation. Our system achieves competitive results on both directions in the final evaluation.

1 Introduction

Automatic post editing (APE) has been used in many scenarios where the performance of a black-box Machine Translation (MT) system is unknown, or, domain specific corrections are required (Pal et al., 2016; Junczys-Dowmunt and Grundkiewicz, 2017; Correia and Martins, 2019; Chatterjee et al., 2020). The continuous improvements of NMT systems's performances along with deep learning advancements insert great challenges on developing sound APE systems, as simple translation errors are rarely seen in machine translation outputs nowadays while the remaining errors are still tough to solve. Transfer learning and data augmentation techniques have demonstrated their efficiency in recent years when models are trained on datasets with limited size (Devlin et al., 2018). Therefore, such techniques are also adopted in APE tasks (Lopes et al., 2019; Chatterjee et al., 2019).

Participants in the APE tasks are required develop systems to automatically post edit the translation outputs from an unknown MT system (Chatterjee et al., 2019). In this year, the dataset has changed in terms of domain (from IT to Wikipedia) and quality of MT (a significant decrease in BLEU). Using previous dataset or officially provided synthetic corpus (such as Artificial and eSCAPE) (Junczys-Dowmunt and Grundkiewicz, 2016; Negri et al., 2018) to enlarge the training set might not be appropriate under such circumstance due to the change in data distribution. Therefore, we decide to perform transfer learning with the officially released training set and integrate Bottleneck Adapter Layers (BAL) (Houlsby et al., 2019; Yang et al., 2020) to prevent over-fitting.

Our model is built based on Transformer (Vaswani et al., 2017) and is pre-trained on the WMT 2019 and 2020 news translation corpora. Compared with the work by (Lopes et al., 2019), we consider that it is more intuitive to use a pre-trained NMT model rather than a pre-trained multilingual language model (LM) (Devlin et al., 2018). During our experiment, we find that fine-tuning the model only on the officially released corpus could easily reach the performance ceiling. As a result, we wondered whether it is possible to introduce external translations as additional MT candidates for data augmentation so as to provide more diversified features. Fortunately, our experiment results demonstrate the effectiveness of such approach. The architecture of our model is shown in Figure 1.

The contributions of our work are as follows:

- We fine-tune the pre-trained NMT models on APE tasks, demonstrating the effectiveness of transfer learning.
- BAL is integrated into the model, further improving the training efficiency as well as the performance.
- Additional MT candidates are introduced to

Proceedings of the 5th Conference on Machine Translation (WMT), pages 797–802
Online, November 19–20, 2020.

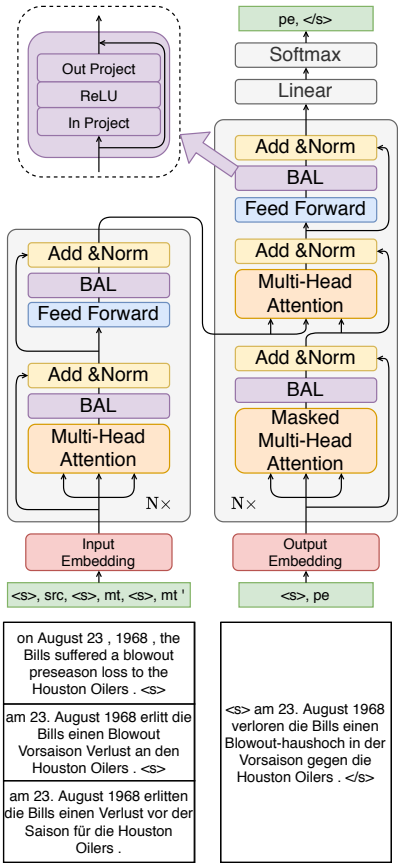


Figure 1: This figure shows the architecture of our model, where MT and augmented MT are concatenated with SRC for passing into the encoder, and PE are generated with the decoder. An example is also shown in the box below the architecture figure.

improve feature diversity, which also significantly improves the performance.

- A detailed case study on the dev set is conducted. We divide post-editing operations into three categories and 10 sub-categories based on their patterns, offering fine-grained suggestions for researchers to build APE models to deal with specific patterns.

2 Task

The dataset contains 7,000 sentences for the training set, 1,000 for the dev and 1,000 for the test. Note that it is also used for the WMT 2020 Word- and Sentence-level Quality Estimation (QE) shared task. Detailed statistics of the dataset is listed in Table 1, showing some metrics of the source (SRC) and translation (MT). From the BLEU (Papineni et al., 2002) scores we can see that the gaps between MTs and PEs are relatively large when compared with that in WMT 2019 (BLEU $>$ 70+).

Attributes	En-De	En-Zh
# Instance	7,000	7,000
# SRC Token	11,4980	115,585
# MT Token	112,342	120,015
% MT Token BAD	28.15	54.33
% MT Gap BAD	4.60	8.04
% SRC Token BAD	26.95	53.60
BLEU (MT, PE)	49.40	30.40
μ(HTER)	0.3181	0.6280
σ(HTER)	0.2017	0.2040

Table 1: The statistics of the training set for both language pairs.

From this aspect, we consider the task is easier this year because over correction will not be a serious problem. However, as the corpora size is much smaller than that of the previous year, this year's APE task is challenging in another way. The evaluation metrics used this year, TER (Snover et al., 2006) and BLEU (Papineni et al., 2002), are exactly the same as that of previous years.

3 Method

3.1 Model

As described in the previous section, due to the limited corpus size, our team decide to employ transfer learning in this task. However, unlike the method proposed in (Lopes et al., 2019), where a multilingual BERT is used as encoder and decoder, we use a pre-trained NMT model and regard it as a more intuitive approach.

We basically treat the APE task as an NMT alike problem, which takes source (SRC) and machine translations (MT) as input and generate PE autoregressively. To adapt this idea with Transformer, we simply concatenate the SRC and MT with the token $<$ s $>$. For models with shared vocabulary and embeddings such as our pre-trained En-De model, this strategy works fine. But for models without sharing input and output embedings such as our En-Zh model, we perform the concatenation with the hidden features after passing through the word embedding separately with the encoder and decoder input embeddings, but keeps the positional embedding normally used.

We perform experiments with this model on the 2019 and 2020 in-domain dataset and find two problems:

- The model converges fast (less than 4 epochs),

but starts over-fitting soon.

- The performance of the model is not good enough on the 2019 dataset, which means it might not be competitive on the 2020 evaluation.

3.2 Bottleneck Adapter Layer

Regarding the first problem, we decide to use the bottleneck adapter layer to reduce the model complexity by only updating the introduced adapter but keeping other parameters fixed. The bottleneck adapter is proposed by (Houlsby et al., 2019), which is similar to the FFN layer in the Transformer but with a low dimensional hidden layer for non-linear activation. In the experiment, we integrate the adapter layer after the self attention layer and the FFN layer for each block in both encoder and decoder. In addition, we find that expanding the hidden size of the neck to the double of the model's hidden size could make the model converge to lower dev loss comparing with using "thinner" or "thicker" necks (i.e. $1/2 \times$ d_model and $2 \times$ d_model). We suppose this size could restrain the complexity of the model at the most suitable level.

3.3 Augmentation with External MT

To further improve the performance, we start investigating the probabilities of adopting external resources for data augmentation. However, as mentioned in previous sections, the domain of eSCAPE (Negri et al., 2018) and the artificial (Junczys-Dowmunt and Grundkiewicz, 2016) corpora are different from that of this task. Afraid of introducing additional biases if incorporating such corpora, we choose to generate more MT candidates (denoted as MT') with the training set and let the model learn complementary information from each other.

More specifically, we first use an additional MT system to create the MT' from the provided SRC text. Then, we simply concatenate the MT' with the SRC and MT sequence to form the new sequence: [SRC, $<$ s $>$, MT, $<$ s $>$, MT$'$], then, use it same as before.

Intuitively, MT' with higher quality can be beneficial for the performance because it is closer to the PE when comparing with the official MT. Therefore, we translate the training set with different MT systems including NMT models trained by us and some publicly available online MT systems.

	En-De		En-Zh	
System	**BLEU**	**TER**	**BLEU**	**TER**
baseline	50.37	31.374	22.62	60.417
+ Fine-tuning	59.51	25.941	31.74	49.257
+ External MT	65.72	20.959	37.37	47.830
+ Ensemble	66.96	20.222	37.83	46.918
Submission	66.89	20.21	37.69	47.36

Table 2: The experimental result of two language pairs evaluated with BLEU and TER on the 2020 dev set, as well as the officially published submission result on the test set. Note that we ensemble 4 and 2 models for En-De and En-Zh, respectively.

Finally, we find that the translation from Google Translate has the best quality (in terms of BLEU for dev set, 67.8 for En-De and 41.77 for En-Zh), and thereby its outputs becomes our augmented MT.

4 Experiment

4.1 Experimental Settings

Our En-De model is implemented with fairseq (Ott et al., 2019) since their published model is pre-trained on WMT 2019 news translation dataset, with BLEU score of 42.7 in evaluation. Our En-Zh model is implemented with THUMT (Zhang et al., 2017) and trained for the WMT 2020 news translation task, which achieved a BLEU score of 46.0 in evaluation. The Transformer model used for both language pairs is Transformer-big with 6 encoders and 6 decoders, and the hidden size is 8192 for FFN layers and 1024 for all other layers.

Note that the vocabulary and encoder/decoder embeddings of the En-De model are shared between two languages and contains 42K of sub-tokens. The vocabulary of the En-Zh model is not shared, and contains 32K and 30K sub-tokens for En and Zh respectively. The BAL used in our model is also modified to have a larger parameter size, where the hidden size of the middle layer is set to 2048.

All models are trained on an Nvidia Tesla V100 GPU with 32G memory. We use the Adam (Kingma and Ba, 2015) optimizer with a constant learning rate of 1e-4 for optimization, and the batch size is 32. FP16 is also used to accelerate training. Models with BALs could converge in less than 8 epochs within 5 minutes.

Categories	Patterns	Num of samples	Proportion	
Knowledge Complement	Named Entity	448	20.38%	38.38%
	Transcreation	206	9.38%	
	Terminology	190	8.65%	
MT Error Correction	Typo	364	16.57%	43.84%
	Disfluency	293	13.34%	
	Illogical	148	6.74%	
	Punctuation Error	71	3.23%	
	Mis-Translation	71	3.23%	
	Over-Translation	16	0.73%	
Stylized Correction	Personal Preference	383	17.45%	17.45%

Table 3: Three categories with eleven types of PE patterns and their proportions, where the MT Error Correction takes the largest part, and are considered as most likely to be solved by APE models.

4.2 Experimental Results

Table 2 shows the experimental results evaluated on the 2020 dev set, where the baseline result is produced by directly calculating scores between the provided MT and PE.

The first experiment is performed by fine-tuning all parameters of the pre-trained Transformer on the official training set, which obtains 8+ of performance gains comparing with the baseline. This demonstrates that fine-tuning the pre-trained NMT model on the limited dataset can be useful.

The experiment of adding external MT for data augmentation shows significant improvements on the performance. However, after performing experiments with different MT candidates, we find that the quality of augmented MTs could influence the performance to a large extent, which motivates us to further improve the robustness of the model.

5 Analysis

Except from focusing on modelling and experimenting, we also conduct an in-depth analysis of the dataset by tagging the PE operations on the dev set. Based on the tags, we categorize PE operations into three categories and try to figure out which kind of PE operations can be learned by an APE system.

We analysis the En-Zh dev set and labeled totally 2196 PE operations, where each sentence has approximately 2.2 corrections. By categorizing these PE operations, we conclude three categories with 10 sub-categories, as described in Table 3. For the first category, SRC text often contains domain specific knowledge or implicit contexts, like Named Entities, terminologies. Strong background knowledge is required when a post-editor translate such text (Yang et al., 2020, 2019). The second category mainly deals with explicit grammar or semantic errors like typo, mis-translations or logical errors, mostly requiring only commonsense to correct. Modifications under the third category are mainly related to the editor's preferences, for example, the format of names and dates. Several examples of the three categories have been shown in the Table 4 in the Appendices.

By observing the output of our system, we find that the first and third categories are relatively difficult for the model to learn in an open domain setting, because of their complexity and uncertainty. For the second category, a pre-trained model has the prior learned from the massive bilingual text, and thereby can be easily fine-tuned to detect and make correction on these mistakes. We believe that further investigation can be performed to explore methods to improve the performance on specific patterns, which is also the research direction of our work.

6 Conclusion

This paper presents our work in the WMT 2020 APE shared task. We adopt transfer learning and data augmentation by fine-tuning a pre-trained Transformer on the provided dataset with external MTs. The experimental results demonstrate the effectiveness of our method. Meanwhile, we achieve competitive results on the test set in the evaluation. Apart from that, we also conducted an in-depth analysis on the dev set, and group the PE operations into several fine-grained categories, serving

as a clearer direction for our future research.

References

Rajen Chatterjee, Christian Federmann, Matteo Negri, and Marco Turchi. 2019. Findings of the WMT 2019 shared task on automatic post-editing. In *Proceedings of the Fourth Conference on Machine Translation, WMT 2019, Florence, Italy, August 1-2, 2019 - Volume 3: Shared Task Papers, Day 2*, pages 11–28.

Rajen Chatterjee, Markus Freitag, Matteo Negri, and Marco Turchi. 2020. Findings of the WMT 2020 Shared Task on Automatic Post-Editing. In *Proceedings of the Fifth Conference on Machine Translation: Shared Task Papers*.

Gonçalo M. Correia and André F. T. Martins. 2019. A simple and effective approach to automatic post-editing with transfer learning. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 3050–3056.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. *arXiv preprint arXiv:1902.00751*.

Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2016. Log-linear combinations of monolingual and bilingual neural machine translation models for automatic post-editing. In *Proceedings of the First Conference on Machine Translation*, pages 751–758, Berlin, Germany. Association for Computational Linguistics.

Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2017. An exploration of neural sequence-to-sequence architectures for automatic post-editing. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing, IJCNLP 2017, Taipei, Taiwan, November 27 - December 1, 2017 - Volume 1: Long Papers*, pages 120–129.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

António V. Lopes, M. Amin Farajian, Gonçalo M. Correia, Jonay Trénous, and André F. T. Martins. 2019. Unbabel's submission to the WMT2019 APE shared task: Bert-based encoder-decoder for automatic post-editing. In *Proceedings of the Fourth Conference on Machine Translation, WMT 2019, Florence, Italy, August 1-2, 2019 - Volume 3: Shared Task Papers, Day 2*, pages 118–123.

Matteo Negri, Marco Turchi, Rajen Chatterjee, and Nicola Bertoldi. 2018. ESCAPE: a large-scale synthetic corpus for automatic post-editing. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, Miyazaki, Japan, May 7-12, 2018*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Santanu Pal, Sudip Kumar Naskar, Mihaela Vela, and Josef van Genabith. 2016. A neural network based approach to automatic post-editing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 2: Short Papers*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA*, pages 311–318.

Matthew Snover, Bonnie J. Dorr, Richard H. Schwartz, and Linnea Micciulla. 2006. A Study of Translation Edit Rate with Targeted Human Annotation.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, \Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

H. Yang, Y. Qin, Y. Deng, and M. Wang. 2020. Nmt enhancement based on knowledge graph mining with pre-trained language model. In *2020 22nd International Conference on Advanced Communication Technology (ICACT)*, pages 185–189.

H. Yang, G. Xie, Y. Qin, and S. Peng. 2019. Domain specific nmt based on knowledge graph embedding and attention. In *2019 21st International Conference on Advanced Communication Technology (ICACT)*, pages 516–521.

Hao Yang, Minghan Wang, Ning Xie, Ying Qin, and Yao Deng. 2020. Efficient transfer learning for quality estimation with bottleneck adapter layer. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation, EAMT 2020, Lisbon, Portugal, 3 - 5 November, 2020*, pages 29–34.

Jiacheng Zhang, Yanzhuo Ding, Shiqi Shen, Yong Cheng, Maosong Sun, Huan-Bo Luan, and Yang Liu. 2017. THUMT: an open source toolkit for neural machine translation. *CoRR*, abs/1706.06415.

A Appendices

Pattern	SRC	MT	PE
Transcreation	12.Bd2 a5 13.Nxc5 bxc5 14.f4 Nd7 15.Bf3 when Jeremy Silman prefers White .	12 . Bd2 a5 13 . Nxc5 bxc5 14 . f4 Nd7 15 . Bf3 , Jeremy Silman 喜欢白色 。	12 . Bd2 a5 13 . Nxc5 bxc5 14 . f4 Nd7 15 . Bf3 , 当Jeremy Silman 掷白棋 时。
Named Entities	however , he finished 2nd in the Budweiser Shootout to Dale Jarrett .	但是，他在布威 赛赛中第二名，以贾雷特而告终。	然而，他在百威啤酒 大赛（Budweiser Shootout ）中获得第二名，仅次于Dale Jarrett 。
Terminology	these include the bald eagle , barn owl , and osprey .	这包括秃鹰 、谷仓猫头鹰 和猎物 .	这些包括秃鹰、仓和鱼鹰 。
Disfluency	Columbia also produced the only slapstick comedies conceived for 3D .	哥伦比亚还制作了为3D 设计的唯一的滑稽喜剧 .	哥伦比亚大学还制作了唯一一部为将会使用3D 技术 播放的滑稽喜剧。
Mis-Translation	although most adult Pacific salmon feed on small fish , shrimp , and squid , sockeye feed on plankton they filter through gill rakers .	虽然大多数的太平洋鲑鱼以小鱼、虾和鱿鱼为饲料，但这些鲑鱼是以浮游生物为饲料的，它们通过刺甲过滤器过滤。	尽管大多数成年太平洋鲑鱼以小鱼、虾和鱿鱼为食，但红鲑以浮游生物为食，它们通过鳃耙过滤。
Over-Translation	’ materials on the Language and Folklore of the Eskimoes , Vol .	” 关于爱斯基摩人的语言和民俗的材料，第二卷 。	爱斯基摩人语言和民俗学材料，卷
Personal Preference	between 1840 and 1890 as many as 40,000 Canary Islanders emigrated to Venezuela .	1840 年至1890 年间，多达40 , 000 加那利群岛居民移居委内瑞拉。	在1840 年至1890 年之间，多达4 万个加那利群岛移民移居到委内瑞拉。

Table 4: This table presents several examples showing the corrections with specific patterns, where the red and green part are the location related to such pattern.

LIMSI @ WMT 2020

Sadaf Abdul Rauf
Univ. Paris-Saclay,
& CNRS, LIMSI

José Carlos Rosales
Univ. Paris-Saclay,
& CNRS, LIMSI
& Inria Paris

Pham Minh Quang
Univ. Paris-Saclay,
& CNRS, LIMSI
& Systran Paris

François Yvon
Univ. Paris-Saclay,
& CNRS, LIMSI

{firstname.lastname}@limsi.fr

Abstract

This paper describes LIMSI's submissions to the translation shared tasks at WMT'20. This year we have focused our efforts on the biomedical translation task, developing a resource-heavy system for the translation of medical abstracts from English into French, using back-translated texts, terminological resources as well as multiple pre-processing pipelines, including pre-trained representations. Systems were also prepared for the robustness task for translating from English into German; for this large-scale task we developed multi-domain, noise-robust, translation systems aim to handle the two test conditions: zero-shot and few-shot domain adaptation.

1 Introduction

This paper describes LIMSI's submissions to the translation shared tasks at WMT'20. This year we have focused our efforts on the biomedical translation task, developing a resource-heavy system for the translation of medical abstract from English into French, using back-translated texts, terminological resources as well as multiple pre-processing pipelines, including pre-trained representations. Systems where also prepared for the robustness task for translating from English into German; for this large-scale task we developed multi-domain, noise-robust, translation systems aim to handle the two test conditions: zero-shot and few-shot domain adaptation.

Machine translation for the biomedical domain is gaining interest owing to the unequivocal significance of medical scientific texts. The vast majority of these texts are published in English and Biomedical MT aims to also make them available in multiple languages. This is a rather challenging task, due to the scope of this domain, and the corresponding large and open vocabulary, including terms and non-lexical forms (for dates, biomedical entities, measures, etc). The quality of the resulting MT output thus varies depending on the amount of biomedical (in-domain) resources available for each target language.

We participated in this years WMT'20 biomedical translation evaluation for English to French direction. English-French is a reasonably resourced language pair with respect to Biomedical parallel corpora, allowing us to train our Neural Machine Translation (NMT) (Sutskever et al., 2014) with only in-domain corpora and dispense with the processing of large out-of-domain data that exist for this language pair. Our main focus for this year's participation was to develop strong baselines by making the best of auxiliary resources: back translation of monolingual data; partial pre-translation of terms; pre-trained multilingual contextual embeddings and IR retrieved in domain corpora. Two pre-prossessing pipelines, one using the standard Moses tools[1] and subword-nmt (Sennrich et al., 2016b) and other using HuggingFace BERT API were developed and compared. All systems are based on the transformer architecture (Vaswani et al., 2017), or and on the related BERT-fused transformer model of Zhu et al. (2020). If our baselines were actually strong, we only managed to get relatively small gains from our auxiliary resources, for reasons that by and large remain to be analyzed in depth. Our biomedical systems are presented in Section 2.

We also participated in the Robustness translation task, developing a multi-domain, noise-robust and amenable to fast adaptation translation system for the translation direction English-German. Our main focus was to study in more depth the adaptor architecture initially introduced in (Bapna and Firat, 2019) in a large-scale setting, where multiple heterogeneous corpora of unbalanced size are available for training, and explore ways to make the system robust to spelling noise in the test data. The zero-shot system is a generic system which

[1] http://www.statmt.org/moses/

Proceedings of the 5th Conference on Machine Translation (WMT), pages 803–812

does not use any adaptation layer; for our few-shot adaptation submission, we did not use the supplementary data provided by the organizers, which turned out to be only mildly relevant for the test condition, but resorted to a data selection strategy. In any case, our submissions are constrained and only use the parallel WMT data for this language pair; they are further described in Section 3.

2 Bio-medical translation from English into French

2.1 Data sources

We trained our baseline systems on a collection of biomedical corpora, *excluding by principle any out-of-domain* parallel corpus, so as to keep the size of our systems moderate and a reduced training time. Table 1 details the corpora used in training.

Parallel			
Corpus	**Wrds (M)**		**Sents.**
	English	**French**	
Ufal	89.5	100.3	2.72 M
Edp	0.04	0.04	2.44 K
Medline titles	5.97	6.43	0.63 M
Medline abstracts	1.23	1.44	0.06 M
Scielo	0.17	0.21	7.84 K
Cochrane-Reference	2.23	2.74	0.12 M
Cochrane-PE	0.43	0.53	20.5 K
Cochrane-GooglePE	0.63	0.77	30.3 K
Taus	20.1	23.2	8.86 M
IR Retrieved	13.2	14.7	3.6M
Development			
Scielo	0.09	0.13	4333
Edp	6.2K	7.1K	328
Khresmoi	28K	33K	1500
Test			
Medline 18	5.7K	6.9K	265
Medline 19	9.8K	12.4K	537
Medline 20	12.7K	16.2K	699
Monolingual			
Corpus	**English** *(Synthetic)*	**French** *(Human)*	**Sent.**
Lissa	8.79	7.70	0.33 M
Med_Fr	16.3	16.2	0.06 M

Table 1: Data sources for the English-French biomedical task (before tokenization)

We gathered parallel and monolingual corpora available for English-French in the biomedical domain. These first included the biomedical texts provided by the WMT'20 organizers: Edp, Medline abstracts and titles (Jimeno Yepes et al., 2017), Scielo (Neves et al., 2016) and the Ufal Medical corpus[2] consisting of Cesta, Ecdc, Emea (OpenSubtitles), PatTR Medical and OpenSubtitles. In addition, we used the Cochrane bilingual parallel corpus (Ive et al., 2016)[3] and the Taus Corona Crisis corpus.[4]. We finally experimented with additional in-domain data selected using Information Retrieval (IR) techniques from general domain corpora including News-Commentary, Books and Wikipedia corpus obtained from Open Parallel Corpus (OPUS) (Tiedemann, 2012). These were selected using the data selection scheme described in (Abdul-Rauf and Schwenk, 2009). Medline titles were used as queries to find the related sentences. We used `3-best` sentences returned from the IR pipeline as additional corpus to build the models (these are shown as X7 in table2).

For development purposes, we used Khresmoi, Edp and Scielo test corpora. The Medline test sets of WMT'18 and 19[5] were used as internal test data.

2.1.1 Monolingual sources

Supplementary French data from two monolingual sources were collected from public archives: abstracts of medical papers published by Elsevier from the Lissa portal[6] and a collection of research articles collected from various sources[7] henceforth referred to as Med_Fr (Maniez, 2009). The former corpus contains 41K abstract and totals approximately 7.7M running words; the latter contains 65K sentences, for a little more than 1.5M running words.

These texts were back-translated (Sennrich et al., 2016a; Burlot and Yvon, 2018) into French using a relatively basic neural French-English engine trained with the official WMT data sources for the biomedical task, using the HuggingFace pipeline (see details below). This system had a BLEU score of 31.2 on Medline 18 test set.

Note that back-translation has also been effec-

[2] https://ufal.mff.cuni.cz/ufal_medical_corpus

[3] https://github.com/fyvo/CochraneTranslations/

[4] https://md.taus.net/corona

[5] With our own sentence alignment.

[6] https://www.lissa.fr/dc/#env=lissa

[7] https://crtt.univ-lyon2.fr/les-corpus-medicaux-du-crtt-613310.kjsp

Symptoms of bacterial pneumonia frequently overlap those present with viral infections or reactive airway disease.
Symptoms of pneumonie bactérienne frequently overlap those present with infections virales or reactive airway maladie.

Figure 1: An example sentence containing pre-translated terms in French

tively used to cater for parallel corpus shortage in the Biomedical domain in (Stojanovski et al., 2019; Peng et al., 2019; Soares and Krallinger, 2019).

2.2 Pre and post-processing

The document level corpora were first retrieved from xml, split[8] into sentences and sentence aligned using Microsoft bilingual aligner (Moore, 2002): these include Cochrane, Scielo and some unaligned documents from Edp. All train, development and test corpora were cleaned by removing instances of empty lines, URLs and lines containing more than 60% non-alphabetic forms.

For tokenization into words and subwords units, two pipelines were considered. The first one is set up as follows (a) tokenize the French and English texts using Moses scripts[9]; (b) compute a joint Byte-pair Encoding (BPE) inventory of 32K units with subword-nmt;[10] (c) generate the translation; (d) detokenize and truecase the output, again with Moses scripts. Systems based on this pipeline are prefixed `M*`. The second one is slightly more complex as it heavily relies on the HuggingFace API[11] for accessing pre-trained BERT models. The corresponding systems are prefixed with `H*` and comprise the following steps: (a) a simple tokenization script, (b) a multilingual segmenter mapping BPE units to pre-trained encodings generated according to (Devlin et al., 2019) as input to the translation system (step (c)). In that case, the MT output is also a sequence of multilingual BPE units that further needs (d) to be reaccentuated and recased, before a final (e) detokenization. Step (d) is non-trivial and is performed by a monolingual translation system trained to convert HuggingFace BPE units into Moses BPE units,[12] which can then be properly reassembled and detokenized as for the Moses pipeline.

2.2.1 Fine-tuning

The fine-tuning process starts from corresponding models trained to convergence, based on BLEU score on dev sets. These are then further fine-tuned using a selected part of the training corpus containing only the Medline abstracts and the three Cochrane corpora, again until convergence. The corresponding systems are post-fixed with `*-ft`.

2.2.2 Pre-translating terms

Medical terms, made of monolexical or polylexical units, are abound in medical terms, and getting their translation right is a very difficult task. Approaches to Biomedical MT have tried to deal this in various ways including explicitly using terminology list (Carrino et al., 2019), domain adaptation (Hira et al., 2019; Stojanovski et al., 2019) and transfer learning (Khan et al., 2018; Peng et al., 2019; Saunders et al., 2019).

We developed systems aimed at improving the translation of terms mainly following the recent proposals of (Dinu et al., 2019; Song et al., 2019). They mostly imply to pre-translate English terms into French, merely replacing the English version with a desired translation in a preprocessing step. The translation system thus inputs mixed-language sentences comprising both English and French words. In our implementation, we followed (Song et al., 2019) and did not mark the pre-translated segments in the input. The target side (French) remained unchanged. Figure 1 displays a sentence extracted from Medline 18 before and after pre-translation (in the latter, French segments are underlined).

Terms are extracted from the French-English version of the Medical Subject Headings thesaurus (MeSH), available in XML format.[13] We extracted a list of about 30K English terms and their preferred translation. This list was extended by searching our training corpus for instances where (a) a term is found in the English sentence; (b) a possible translation is found in the French sentence. Step (b)

[8] https://github.com/berkmancenter/mediacloud-sentence-splitter

[9] http://www.statmt.org/moses/

[10] https://github.com/rsennrich/subword-nmt

[11] https://Huggingface.co/transformers/model_doc/bert.html

[12] This process is not completely error prone, and yields a BLEU score of 98.2 on Medline 18 test set.

[13] http://mesh.inserm.fr/FrenchMesh/

ID	Train	Detail	ID	Medline			ID	Medline			ID	Medline		
				18	19	20		18	19	20		18	19	20
				Moses				HuggingFace						
X0	**wmt**	WMT data	**M0**	20.7	22.6	27.3	**H0**	26.8	29.6	33.7	**B0**	26.1	29.0	32.9
X1	**base**	All data	**M1**	24.7	25.9	32.6	**H1**	27.7	30.2	35.9	**B1**	28.6	31.1	37.2
X2	**base-ft**	X1 ⇒ X2	**M2**^{*2}	25.6	26.1	32.9	**H2**	28.1	30.0	35.5	**B2**	38.8	29.5	35.8
Back Translations of Monolingual data														
X3	**base+bt**	X1 + BT	**-**	-	-	-	**H3**	27.9	30.8	36.7	**B3**	28.0	31.0	36.3
X4	**base+bt-ft**	X3 ⇒ X4	**-**	-	-	-	**H4**^{*1}	28.7	30.7	37.0	**B4**	31.6	30.8	36.2
Using Pre-translated terms														
X5	**base+bt-pt**	X3 ⇒ X5	**-**	-	-	-	**H5**	27.5	30.0	35.9	**B5**	29.0	30.2	36.3
X6	**base+bt-pt-ft**	X5 ⇒ X6	**-**	-	-	-	**H6**	33.0	27.0	32.5	**B6**	36.0	28.8	35.2
Using IR retrieved corpus														
X7	**base+bt+IR**	X3 + IR	**-**	-	-	-	**H7**	28.8	31.4	37.2	**B7**	28.8	31.2	36.5
X8	**base+bt+IR-ft**	X7 ⇒ X8	**-**	-	-	-	**H8**	29.4	31.0	37.3	**B8**^{*3}	31.7	30.6	36.5

Table 2: BLEU scores for the various biomedical systems on Medline 18, 19 and 20 test sets. Superscripts ^{*n} denote the runs submitted: H4, M2, B8.

relies on a much larger list of about 800K possible associations, also extracted from the MeSH. The final term list contains about 40K entries.

Training was performed in two steps: starting with our best system (M3), we resume training with partially pre-translated sentences, using only the following corpora: Cochrane, Medline, Taus and a large portion of Scielo (for a grand total of 2M sentence pairs). This process is performed until convergence. The same fine-tuning process as described above is optionally performed.

In testing, we replace any matching English term with its translation subject to length constraints to avoid irrelevant, ambiguous or accidental matches. We only substitute terms of (source+target) length greater or equal to 7 characters, yielding the pre-translation of 462 and 795 terms respectively in the Medline 18 and Medline 19 test sets. Cases where one term has several translations are disambiguated based on frequency of occurrences in training. These systems appear in the last two rows of Table 2 with the postfix `*-pt`.

2.3 Translation framework

We mostly used two architectures to build our systems: basic Transformer models (Vaswani et al., 2017) as well as BERT-fused transformer models (Zhu et al., 2020). All systems use Facebook's seq-2-seq library fairseq (Ott et al., 2019) with parameters settings borrowed from `transformer_iwslt_de_en`.[14] We used memory efficient FP16 optimizer. The ReLU activation function was used in all 6 encoder and 6 decoder layers, 1024 hidden layer size and batch size of 4K. Training was optimized using Adam and a learning rate of 0.0005 was fixed for all experiments.

For the BERT-based models, we relied on BERT-NMT.[15] This allowed us to build the BERT-fused models using the same architecture and parameters as the baseline transformer models and to establish fair comparisons. In BERT-fused NMT model, the contextual representations are first computed by the BERT model for each token (in the source and target), these are then combined at each encoder and decoder layer using the attention mechanism. Full details are in Zhu et al. (2020).

Given the size of our training data, the "lazy" output dataset implementation was used to enable data loading in the RAM. Systems were trained until convergence based on the BLEU score on the development sets. Evaluation is performed using sacrebleu (Post, 2018). Scores are chosen based on the best score on the development set (Khres+Edp+Scielo) and the corresponding scores for that checkpoint are reported on Medline 18 and

[14] https://fairseq.readthedocs.io/en/latest/models.html

[15] https://github.com/bert-nmt/bert-nmt

Medline 19 test sets. For systems using terminology pre-translation, Khresmoi and Edp were used as development sets.

2.4 Results

Results are in Table 2, where we report BLEU scores for the three tracks explored in this work. $M*$ denotes the Moses tokenization pipeline, $H*$ represents the HuggingFace pipeline and $B*$ denotes the BERT models with HuggingFace tokenization. We computed the scores on Medline 18, Medline 19 and Medline 20 test sets,[16] based on the best checkpoint on our development corpus. Base systems are given on the left, ($\Rightarrow$) identifies the derived (fine-tuned) systems.

We first built baseline systems for the three tracks. `X0` denotes the systems built using only the data provided by the organizers. `X1` are our baseline systems built using all our parallel corpora. We see a unanimous improvement in all tracks ranging from 0.6 to 5.3 BLEU points, which is obtained by adding around $1M$ sentences of additional Cochrane and Taus corpora to the already available $2.9M$ sentences from WMT20. This hints at the relevance of the additional in-domain parallel corpora used.

These baselines `X1` are then further fine-tuned with Cochrane and Medline abstracts as discussed in section 2.2.1, these are shown post-fixed with `*-ft`. All the systems show an improvement in the Moses track. Similarly, we see gain for all tracks for Medline 18 with the highest improvement on BERT-fused systems. For Medline 19 and 20, fine-tuning resulted in a small drop in performance across the board (except than Moses track), for reasons that remain to be analyzed.

Comparing M1-M2 with H1-H2, we see that the Moses pre-processing, which is simpler that HuggingFace's and relies on domain-adapted BPE units is slightly better than the alternative. As using HuggingFace's tools was a way to also experiment with BERT and other extensions, it was nonetheless used for the other systems.

Having established the adequacy of the supplementary parallel corpora, we built systems with back-translated monolingual corpora (section 2.1.1). These appear as X3 and X4 in Table 2. These back-translations were somewhat helpful, not to the extent that we were expecting them to be. Comparing with our baseline X1 systems, we see a small gain of (0.2,0.6,0.8) for our transformer models using HuggingFace tokenization (H1 vs. H3) but no gain for the BERT track (B1 vs. B3). We can speculate about various reasons for this behaviour: (a) genre mismatch with the test set: even though the monolingual corpora also contain scientific texts in biomedical domain, the use of full documents might yield subtle differences in style and term used with what is observed in abstracts, which are more rigidly structured; (b) the use of a comparatively small amount of back-translations as compared to the baseline corpora; (c) the quality of back translations.

Our experiments with pre-translated terms resulted in a small drop of the BLEU scores for the corresponding systems (X5, X6). Our initial analysis of term use[17] in the references and in the system outputs helps understand why this is the case. As it turns out, references translations contain a smaller proportion of *licensed terms* than our baseline translations (55.6% for the reference, 61.1% and 61.6% for respectively X3 and X4), which in turn contain less terms than our term-sensitive systems (H5 and H6, for which these numbers are respectively 68.9 and 64.2). Another way to look at this is to realize that only 58.6% of our pre-translations were actually in the reference. All in all, using more translations from the MeSH makes our output less similar to the reference than the baselines, and contributes to degrade the BLEU score. It is however reinsuring to see that pre-translating terms actually increases the number of terms in the output – in fact, for H5 and H6 we find that respectively 84.2% and 81.9% of these pre-translations are actually copied in the target, even though there was no indication of these French inserts in the mixed-language input. We can also note that the majority of the pre-translated terms were frequent Biomedical terms (such as "patients", "health", etc) that were also correctly translated by the baseline systems. Evaluating these outputs with more useful metrics than BLEU still needs to be performed.

Adding the IR retrieved sentences finally brought us nearly one extra BLEU point on all test sets for the HuggingFace systems, but not much improvement for the BERT-fused system.

[16] Again with our own sentence alignment.

[17] Based on the proportion of source word in our term list that are actually translated with a translation that exists in the Mesh. These proportions are computed on an aggregate of the Medline testsets for 2018, 2019 and 2020, only counting terms with source+target length greater than 7.

Domain	Corpus	sents.	words (en)	words (de)
web	Paracrawl	50,875	978	919
economy	Tilde EESC	2,858	61	58
news	Commoncrawl	2,399	51	47
	Tilde rapid	940	20	19
	News commentary	361	8	8
tourism	Tilde tourism	7	0.1	0.1
gov	Epps	1,828	45	42
medical	Tilde EMEA	347	5	5
banking	Tilde ECB	4	0.085	0.074
wiki	Wikipedia Matrix	5,473	91	88

Table 3: Data used in the Robustness task: number of parallel lines ($\times 10^3$), number of tokens ($\times 10^6$)

2.5 Conclusion

In conclusion, our participation to this year's WMT biomedical task has enabled us to develop basic tools and pipelines for a variety of architectures and to start exploring domain-adapted extensions of a baseline Transformer architecture, using complementary resources, such as supplementary corpora, pre-trained embeddings and terminological resources. If all these extensions were not equally useful, we still were able to develop strong systems for this task that provide us with a solid starting point for further developments of domain-adapted NMT systems.

3 Robustness: translating English challenge test sets into German

3.1 Data sources

Our sole data sources are the parallel corpora distributed by the organizers for the News task, which we significantly down-sampled in order to reduce the overall computational training cost. Monolingual data sources were not considered. These parallel corpora were then grouped into 8 broad domains. Statistics for each corpus / domain are in Table 3.

Our development set is composed of a varied set of common benchmarks, aimed to represent a wide diversity of genres and domains.

3.2 Pre-processing

The first step of pre-processing consists of cleaning the parallel corpora using the following rules: (a) discard sentences based on length (with a maximum length of 99 words), and on the source/target length ratio (in the interval $[2/3; 3/2]$); (b) discard instances of non-English and non-German sentences, using the langid toolkit;[18] (c) remove duplicates sentence pairs. After cleaning, the parallel corpus used in training contains 50,875,449 sentences pairs.

The next step is to lowercase and to tokenize the text into words and subword units. We use the Tokenizer library from OpenNMT.[19] We first lowercased every word, adding a special marker at the beginning of capitalized words, and likewise for uppercased words and segments. For instance, this procedure replaces "It" with "**U** it", and "NOVEMBER RAIN" with "**BU** november **EU BU** rain **EU**". These markers are preserved during the BPE tokenization. We learned a joint BPE vocabulary for both languages using 32K merge operations.

3.3 Training a robust multi-domain system

Our approach to robustness aims at building a system that (a) could fare well for test sets that would be similar to the training domain; (b) could also accommodate data from new, unseen, domains; (c) would be easy to adapt to a new domain (for the few-shot condition); (d) could be robust to spelling noise in the test. Requirements (a)-(c) lead us to implement an extension of the baseline Transformer architecture with residual adapters (more on this in section 3.3.2); to meet requirement (d), we implemented a data augmentation technique described in Section 3.3.3.

3.3.1 Baseline

The baseline system relies on the Transformer Large architecture from (Vaswani et al., 2017). We set the embeddings size and the hidden layers size to 1024. Transformers use multi-head attention with 16 heads in each of the 6+6 layers; the inner feedforward layer contains 4096 cells. Training uses a batch size of 12288 tokens; optimization uses Adam with parameters $\beta_1 = 0.9$, $\beta_2 = 0.98$ and Noam decay ($warmup\_steps = 4000$) and a dropout rate of 0.1 for all layers.

3.3.2 Residual adapters

Our main source of inspiration is the work of Bapna and Firat (2019), who initially introduced the use of residual adapter modules for domain adaption. In a nutshell, this proposal adds an additional, domain-specific layer on top of every layer of the encoder and the decoder. It thus provides us with

[18] https://github.com/saffsd/langid.py
[19] https://github.com/OpenNMT/Tokenizer

a lightweight, computationally efficient alternative to domain adaptation with full fine-tuning, which implies to update all the system parameters. We generalize this approach by training (or rather fine-tuning) a distinct residual adapter for each of the 8 train domains, while freezing the parameters of the baseline (generic) system. These adapter modules are made of 2-layer perceptrons, with an inner ReLU activation function operating on normalized entries of dimension 2048.

Any test sentence from a known domain would then use the corresponding adapter; for test sentences from new domains two options are possible: use only the generic system (without adapter), or use the adapter for the more similar domain. This methodology was chosen in the view of the few-shot task, where a new adapter could easily be learned for a new domain, even with a very small amount of data.

We evaluate the effectiveness of the residual adapters architecture using a varied set of internal test sets. Table 4 reports the BLEU scores of the baseline, generic model, prior to adaptation, as well as the adapted system. As expected, performance are overall better when selecting the appropriate domain for each test set.

We applied this idea to improve the ability our generic model to handle noisy data. Recall that most of the training data (with the exception of the web domain) comes from "clean" sources. To this end, we generated artificial training data for an additional "noise" domain, by automatically altering the source side of randomly selected training data. The noise generation procedure is described below. By doing this way, we expect the model to take advantage of the residual layer when input with noisy data that is similar to our artificial noisy domain, while keeping (a) its good performance on the other known domains, (b) a reasonable behaviour on any other clean data (using the generic baseline model without adapter).

3.3.3 Artificial noise generation

In order to account for possible user generated content (UGC) at test time, we explored the possibility of learning typical UGC noise at the character-level. To this end, we used an automatically scrapped Wikipedia correction corpus (Grundkiewicz and Junczys-Dowmunt, 2014), which has been filtered to keep only word replacements with, at most, a character edit distance of 30% of the word length. In the end, we kept a total of roughly 17.8M pairs of errors and editions. We then trained a character-level Transformer with the same architecture as our base translation model, which had a perfect-match error rate of 22% on the test data partition. Finally, we augmented the original training data by sampling random original words according to a uniform probability distribution and replacing them with the prediction of our character-based UGC noise generator, resulting in the same number of sentences in the original corpora. We have set a 7% probability of replacement, that has been estimated by the percentage of Out-of-Vocabulary words in a real-world UGC corpus. This heuristic later seemed, as discussed in Section 3.4, to overestimate the quantity of noise to be added and, in retrospective, we should have used other metrics to estimate the noise level, such as the n-gram Kullback-Leibler divergence, as discussed in (Alonso et al., 2016; Rosales Núñez et al., 2019). Table 5 displays some examples of noise entries produced by our character-based generator. Regarding these, although typographical errors prevail, due to the nature of automatic filtering of the Wikipedia editions, some learned replacements operations can change the semantics and syntax of the sentence, e.g. (using $\rightarrow$ use), (for $\rightarrow$ in) or (may $\rightarrow$ can); thus introducing unexpected confusion in the training data.

3.4 Results

We report the BLEU scores of our various systems in Table 6. Our submission to the zero-shot evaluation was `FT-Adapt-Noise`, which we found was sub-optimal afterwards. However, interestingly, the residual adapter mechanism proved to substantially outperform the classical fine-tuning of the whole model (i.e. `FT-Full-Noise`). Finally, the residual adapter fine-tuned using the ParaCrawl corpus (`FT-Adapt-Web`) had the best performance on the test set, probably due to the higher similarity of this corpus to the target test. In addition, we noted that the baseline and `FT-Adapt-Noise` output a considerable number of English phrases, leaving most of the source sentence unchanged, whereas the `FT-Adapt-Web` reduced the number of sentences that presented this issue.

In order to assess how much the 172 sentences that were left completely untranslated impact the performance of the `FT-Adapt-Noise` model, we replaced them with the output of the

Test set Domain	IT tech	Khresmoi medical	NT17	NT18 news	NT19	EPPS gov	EESC eco	RAPID news	Tourism tourism	Wiki wiki	ECB bank
Baseline	36.27	29.78	26.24	41.27	37.24	29.31	30.48	31.93	17.64	14.92	38.11
FT-Adapt domain	-	29.46	26.48	41.43	37.24	29.65	30.45	32.43	19.21	-	48.99

Table 4: BLEU scores on various test sets using our baseline and adapted NMT systems for each domain. *NT stands for NewsTest*

original	the	combination	may	concerning	using	no	common	developing	for	status	also
noisy	this	combonation	can	concering	use	not	comon	developping	in	staus	aslo

Table 5: Examples of clean and artificially noisy word inputs

`baseline` and observed a performance increase to 31.3 BLEU. This suggests that our data augmentation technique introduced confusion to the base model after fine-tuning and the resulting translation system was less adapted to the zero-shot test set.

	robustness-set1	#EN Sents.
Baseline	31.6	120
FT-Adapt-Noise	30.2	172
FT-Full-Noise	24.6	256
FT-Adapt-Web	34.2	34
FT-Full-Web	33.8	49

Table 6: BLEU scores for the EN-De models developed for the Robustness track. We also report, for each system, the number of sentences that were left unchanged.

The design and organization of the few-shot part of the evaluation was not fully satisfactory: while we did train an adapter module using the new data seemingly corresponding to a novel domain, it seems that the corresponding test set was never released and we could not fully evaluate our approach. Working on this task was nonetheless very instructive, and helped us better understand the strength and pitfall of the residual adapter architecture when applied to a very large scale task and in the face of unbalanced, heterogeneous, training data.

4 Conclusions

In this paper, we have described the development undertaken for this year's participation to WMT shared tasks. Taking part to the Biomedical track as allowed us to collect and prepare useful resources (monolingual and bilingual corpora, term lists) for this domain, and to explore several pipelines and translation architectures. The general results are overall satisfactory, even though a deeper analysis of the MT is still needed to strengthen our conclusions. They will also help us prepare for next year tasks, where we expect to work on more language pairs. Our experiment for the Robustness track were less successful: we were not really prepared for the general tone and style that was observed in the zero-shot test set; we also did not understand the general orientation taken for the few-shot adaptation, as it seemed to us that the adaptation data was not really relevant for the only test set that was ever released.

Acknowledgments

This work is (partly) based on computations performed on the Saclay-IA and on the Jean ZAY computing platforms. The authors wish to thank Pierre Zweigenbaum for his help finding French corpora in the biomedical domain and Hicham El-Boukkouri for providing guidance setting up BERT-based systems. The second author wishes to acknowledge the help and guidance of Djamé Seddah and Guillaume Wisniewski; his work is funded by the French Research Agency via the ANR project ParSiTi (ANR-16-CE33- 0021).

References

Sadaf Abdul-Rauf and Holger Schwenk. 2009. On the use of comparable corpora to improve SMT performance. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 16–23, Athens, Greece. Association for Computational Linguistics.

Héctor Martínez Alonso, Djamé Seddah, and Benoît Sagot. 2016. From noisy questions to minecraft texts: Annotation challenges in extreme syntax scenario. In *Proceedings of the 2nd Workshop on Noisy*

User-generated Text, NUT@COLING 2016, Osaka, Japan, December 11, 2016, pages 13–23. The COLING 2016 Organizing Committee.

Ankur Bapna and Orhan Firat. 2019. Simple, scalable adaptation for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, EMNLP-IJCNLP, pages 1538–1548, Hong Kong, China. Association for Computational Linguistics.

Franck Burlot and François Yvon. 2018. Using monolingual data in neural machine translation: a systematic study. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 144–155, Brussels, Belgium. Association for Computational Linguistics.

Casimiro Pio Carrino, Bardia Rafieian, Marta R. Costa-jussà, and José A. R. Fonollosa. 2019. Terminology-aware segmentation and domain feature for the WMT19 biomedical translation task. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 151–155, Florence, Italy. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Georgiana Dinu, Prashant Mathur, Marcello Federico, and Yaser Al-Onaizan. 2019. Training neural machine translation to apply terminology constraints. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3063–3068, Florence, Italy. Association for Computational Linguistics.

Roman Grundkiewicz and Marcin Junczys-Dowmunt. 2014. The wiked error corpus: A corpus of corrective wikipedia edits and its application to grammatical error correction. In *Advances in Natural Language Processing – Lecture Notes in Computer Science*, volume 8686, pages 478–490. Springer.

Noor-e Hira, Sadaf Abdul Rauf, Kiran Kiani, Ammara Zafar, and Raheel Nawaz. 2019. Exploring transfer learning and domain data selection for the biomedical translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 156–163, Florence, Italy. Association for Computational Linguistics.

Julia Ive, Aurélien Max, François Yvon, and Philippe Ravaud. 2016. Diagnosing high-quality statistical machine translation using traces of post-edition operations. In *International Conference on Language Resources and Evaluation - Workshop on Translation Evaluation: From Fragmented Tools and Data Sets to an Integrated Ecosystem (MT Eval 2016 2016)*, page 8, Portorož, Slovenia.

Antonio Jimeno Yepes, Aurélie Névéol, Mariana Neves, Karin Verspoor, Ondřej Bojar, Arthur Boyer, Cristian Grozea, Barry Haddow, Madeleine Kittner, Yvonne Lichtblau, Pavel Pecina, Roland Roller, Rudolf Rosa, Amy Siu, Philippe Thomas, and Saskia Trescher. 2017. Findings of the WMT 2017 biomedical translation shared task. In *Proceedings of the Second Conference on Machine Translation*, pages 234–247, Copenhagen, Denmark. Association for Computational Linguistics.

Abdul Khan, Subhadarshi Panda, Jia Xu, and Lampros Flokas. 2018. Hunter NMT system for WMT18 biomedical translation task: Transfer learning in neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 655–661, Belgium, Brussels. Association for Computational Linguistics.

François Maniez. 2009. L'adjectif dénominal en langue de spécialité: étude du domaine de la médecine. *Revue française de linguistique appliquée*, 14(2):117–130.

Robert C. Moore. 2002. Fast and accurate sentence alignment of bilingual corpora. In *Proc. AMTA'02*, Lecture Notes in Computer Science 2499, pages 135–144, Tiburon, CA, USA. Springer Verlag.

Mariana Neves, Antonio Jimeno Yepes, and Aurélie Névéol. 2016. The Scielo Corpus: a parallel corpus of scientific publications for biomedicine. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 2942–2948, Portorož, Slovenia. European Language Resources Association (ELRA).

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Wei Peng, Jianfeng Liu, Liangyou Li, and Qun Liu. 2019. Huawei's NMT systems for the WMT 2019 biomedical translation task. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, Florence, Italy. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

José Carlos Rosales Núñez, Djamé Seddah, and Guillaume Wisniewski. 2019. Comparison between NMT and PBSMT performance for translating noisy user-generated content. In *Proceedings of the 22nd Nordic Conference on Computational Linguistics*, pages 2–14, Turku, Finland. Linköping University Electronic Press.

Danielle Saunders, Felix Stahlberg, and Bill Byrne. 2019. UCAM biomedical translation at WMT19: Transfer learning multi-domain ensembles. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 169–174, Florence, Italy. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany.

Felipe Soares and Martin Krallinger. 2019. BSC participation in the WMT translation of biomedical abstracts. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 175–178, Florence, Italy. Association for Computational Linguistics.

Kai Song, Yue Zhang, Heng Yu, Weihua Luo, Kun Wang, and Min Zhang. 2019. Code-switching for enhancing NMT with pre-specified translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 449–459, Minneapolis, Minnesota. Association for Computational Linguistics.

Dario Stojanovski, Viktor Hangya, Matthias Huck, and Alexander Fraser. 2019. The LMU munich unsupervised machine translation system for WMT19. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 393–399, Florence, Italy. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3104–3112.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eight International Conference on Language Resources and Evaluation*, LREC'12, Istanbul, Turkey. European Language Resources Association (ELRA).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Jinhua Zhu, Yingce Xia, Lijun Wu, Di He, Tao Qin, Wengang Zhou, Houqiang Li, and Tieyan Liu. 2020. Incorporating BERT into Neural Machine Translation. In *Proceedings of the International Conference on Learning Representations*, ICLR.

Elhuyar submission to the Biomedical Translation Task 2020 on terminology and abstracts translation

Ander Corral
Elhuyar Foundation
a.corral@elhuyar.eus

Xabier Saralegi
Elhuyar Foundation
x.saralegi@elhuyar.eus

Abstract

This article describes the systems submitted by Elhuyar to the 2020 Biomedical Translation Shared Task, specifically the systems presented in the subtasks of terminology translation for English-Basque and abstract translation for English-Basque and English-Spanish. In all cases a Transformer architecture was chosen and we studied different strategies to combine open domain data with biomedical domain data for building the training corpora. For the English-Basque pair, given the scarcity of parallel corpora in the biomedical domain, we set out to create domain training data in a synthetic way. The systems presented in the terminology and abstract translation subtasks for the English-Basque language pair ranked first in their respective tasks among four participants, achieving 0.78 accuracy for terminology translation and a BLEU of 0.1279 for the translation of abstracts. In the abstract translation task for the English-Spanish pair our team ranked second (BLEU=0.4498) in the case of OK sentences.

1 Introduction

General purpose translation systems usually perform poorly where domain specific knowledge is required (Koehn and Knowles, 2017). Therefore, it is essential to develop translation systems for specific domains. However, high quality parallel corpora for domain specific tasks are only available for a few major languages and obtaining in domain translated data for the majority of language pairs becomes a challenge. With such scarcity of resources, various domain adaptation techniques have shown promising results (Currey et al. 2017, Saunders et al. 2019, Sennrich et al. 2017).

The biomedical domain is of great interest for the application of machine translation. It is a sector of great importance in society, even more so after the advent of COVID-19, where the handling of documentary information plays a major role. Therefore, it is a scenario where machine translation can be of great help in order to facilitate the flow of information between different languages.

We can find different works in the literature that address the task of developing NMT systems adapted to the biomedical domain (Yepes et al., 2017; Sennrich et al., 2017; Khan et al., 2018). In contrast, few papers focus on language pairs where training data is scarce. This is precisely the case of the task of translation from English to Basque. The work of Soto et al. (2019) is specially relevant in this case, which presents a clinical domain oriented system for Basque-Spanish based on RNN and Transformer architectures and which does not require bilingual domain texts but only bilingual clinical terminology (SNOMED CT).

Our participation in the Biomedical Translation Shared Task addresses the translation of biomedical terminology from English to Basque and the translation of biomedical abstracts from English to Basque and from English to Spanish. Given the scarcity of real parallel biomedical domain corpora, our approach focused on different strategies to combine open domain data with in-domain biomedical data. In the case of the English to Basque translations, back translation technique has been applied to generate synthetic examples from a real biomedical Spanish-Basque corpus and several monoligual in-domain Basque corpora have been merged to the training data by copying target side examples to source. Furthermore, we have considered finetuning previously available open domain systems in order to take advantage of the learnt general, open domain patterns.

This article is structured as follows: the following section reviews the most relevant related works. Section 3 describes the systems presented to the sub-tasks that include the English-Basque and English-Spanish pairs, as well as the results

Proceedings of the 5th Conference on Machine Translation (WMT), pages 813–819
Online, November 19–20, 2020.

obtained in the experimentation. Section 4 presents the results obtained in the official evaluation, and finally, we end the paper outlining the most relevant conclusions drawn from this work.

2 Related work

Most of the related work on domain adaptation focuses on using either in-domain monolingual corpora, synthetically generated corpora or small parallel corpora.

Regarding in-domain monoligual corpora, Currey et al. (2017) analyze the benefits of augmenting available data by copying the target side monolingual data to source and training the system. Results show significant gains in accuracy on named entities and words remaining identical in source and target languages.

Several studies show the effectiveness of generating synthetic parallel corpora for domain adaptation. Sennrich et al. (2015) use the back translation technique with target monolingual data to strengthen the decoder, Zhang and Zong (2016) make use of source side monolingual data and Park et al. (2017) use both, source and target monolingual data to improve in-domain translations.

Finally, when in-domain parallel corpora is available, previous work focuses on mixed domain NMT systems by using both in-domain and out of domain data. Chu et al. (2017) propose to use control tags to mark in-domain sentences prior to concatenating multiple domain corpora. Sajjad et al. (2017) compare different methods for training a multi domain system, such as, concatenation, interactively training on different domains, selecting out of domain data close to the in domain data and ensembling different domain models. Wang et al. (2017) exploit the internal sentence embeddings to find sentences that are close to in-domain data from out of domain data.

Several works in the literature address the task of developing NMT systems adapted to the biomedical domain. Saunders et al. (2019) apply transfer learning technique by training on a large, general domain corpus and finetuning a series of systems on different biomedical domains. They perform multi domain ensembling to further improve the results. Khan et al. (2018) iteratively apply transfer learning on various biomedical domains.

Regarding works dealing with Basque the work of Soto et al. (2019) presents a clinical domain oriented system for Basque-Spanish based on RNN and Transformer architectures, which does not require bilingual domain texts but bilingual clinical terminology (SNOMED CT). They also analyse different back translation techniques. Aimed at translating clinical terminology into Basque, we find the work of Perez-de Viñaspre (2017) which proposes a system for translating SNOMED CT into Basque by combining lexical resources, transliteration of neoclassical terms, generation of nested terms and a domain-adapted RBMT system.

3 Experiments

In this section we describe the experiments carried out when training the translation systems. We compare the results obtained by the systems on different open domain and in-domain test sets prior to selecting the best runs to submit for the biomedical translation task. We also detail the Transformer architecture used and its parameters.

3.1 Datasets

For the experiments, we considered open domain general data and in-domain task specific data for fine-tuning purposes. Open domain data comprises the publicly available Paracrawl v5 (Esplà et al., 2019) corpus for English-Spanish and a Elhuyar's internal synthetic corpus for Basque-Spanish.

Due to the scarcity of in-domain data in the case of English-Basque, back translation has also been applied to generate synthetic examples. Furthermore, we augmented training data by using monolingual Basque data gathered from artificially generated hospital notes, SNOMED-CT terminological content and Wikipedia biomedical articles. We have created parallel data by copying the Basque sentences to source so that each source sentence is identical to the target sentence. Table 1 offers a summary of the corpora used.

3.2 Architecture

For training the models the Transformer architecture (Vaswani et al., 2017) has been chosen. Specifically, the Python implementation of the OpenNMT (Klein et al., 2017) library has been used. Transformers are based on an encoder-decoder system with an attention mechanism. Both the encoder and the decoder are composed of 6 layers composed in turn by a feed forward network and a multi-head attention mechanism. Default values of the architecture without any optimization of the parameters have been applied. The size of the recurrent neu-

Corpus	Pair	Domain	Description	Sentences
Elhuyar synthetic (ELH Syn)	EN-EU	Open	Elhuyar's internal synthetic corpus obtained by translating an internal ES-EU corpus with our best performing ES-EN out of domain system	6.9M
EHU books	EN-EU	Biomed.	Collection of biomedical books translated from English to Basque	22.6k
ICD-10	EN-EU	Biomed.	ICD-10 codes translated from English to Basque and publicly available for the shared task	25.9k
EHU Synthetic (EHU Syn)	EN-EU	Biomed.	Synthetically generated corpus by back translating an in domain EHU book collection dataset (ES-EU) from Basque to English with a previously trained in domain EU-EN system	303k
Hospital notes	EU	Biomed.	Artificially generated hospital notes to use as guide for practitioners	2.2k
SNOMED	EU	Biomed.	Automatic translation of the terminological content of SNOMED-CT (2020), no manually revised	105k
Wikipedia	EU	Biomed.	Medical domain articles from Wikipedia	1.3k
Paracrawl v5 (PCv5)	EN-ES	Open	Publicly available Paracrawl v5 corpus, which comprises parallel segments crawled from the web	33.3M
Biomedical (Mix)	EN-ES	Biomed.	Comprises subsets of the Scielo dataset (Soares et al., 2018), and previous years' WMT shared task datasets	560k

Table 1: Size of corpora used for training the systems.

ral network of each layer is 512. Thus, 512 size embeddings have been used for both source and target sentences. Adam optimizer has been used during the training, and a learning-rate of 2 with a warm-up phase of 8000 steps. The dropout ratio is 0.1 and the batch size is 4096 sentences. All models have been trained until the results on the development set stopped improving.

To avoid the open vocabulary issue and for a better translation of unknown words, BPE tokenization (Sennrich et al., 2016) has been applied to source and target sequences. Rare or unseen words are represented as a sequence of subword units. In the case of Basque, this encoding is particularly useful as declensions generate a larger vocabulary.

3.3 EN-EU experiments

In this section we provide a detailed description of the experiments carried out for the English-Basque pair. We describe the systems built and the results obtained on different test sets. Looking for a robust experimental setup, we conducted both open domain and in-domain evaluation. A brief description of the test sets can be found on Table 2.

3.3.1 Systems

For the English-Basque pair we trained the following systems:

Baseline1. A strong baseline by pivoting Elhuyar's best out of domain EN-ES and ES-EU models. These models are trained on the PCv5 corpus and a Elhuyar's internal Spanish-Basque corpus respectively.

Baseline2. A further improvement of Baseline1, by fine-tuning the EN-ES pivoting system with Medline and SCIELO in-domain biomedical data.

Baseline3. A previously available out of domain EN-EU system trained with back translated synthetic data.

Baseline4. A simple baseline trained with the task's official in-domain data (ICD-10 corpus).

SystemA1, SystemA2 and SystemA3. These systems are the result of fine-tuning Baseline3 with in-domain data. SystemA1 uses a subset (250k) of ELH Syn open domain corpus as well as shared task ICD-10 in domain data. A small portion of the ICD-10 corpus (1k) is used for validation. SystemA2 is a variant of SystemA1 by adding more in domain data from the EHU books corpus. A small subset of the EHU books corpus (1k) is also added to the validation set. Finally, SystemA3 includes a subset (5k) of the out of domain ELH Syn corpus in the validation set in order to prevent the system from forgetting about prior out of domain knowledge.

SystemA4. A variant of SystemA3 using all the available ELH Syn open domain data.

SystemB1. This system was trained by adding synthetically generated EHU Syn in-domain data to the data used in SystemA4. To create synthetic data, a fine-tuned EU-EN model has been used to back translate an internal ES-EU biomedical corpus gathered from a collection of EHU books.

SystemB2. This system was trained by further augmenting data from SystemB1 with copied monolingual Basque target data from the shared task (Hospital notes, SNOMED terminology and Wikipedia).

SystemC. A variant of SystemB2. Synthetic Medline data from the WMT19 EN-ES biomedical shared task was added to the validation set to improve the performance on the Medline domain.

Test	Pair	Domain	Description	Sentences
Synthetic (Syn)	EN-EU	Open	Subset of the Elhuyar's internal synthetic corpus	5k
EHU books (EHU)	EN-EU	Biomed.	Subset of the collection of biomedical books translated from English to Basque	1k
Medline pro (PRO)	EN-EU	Biomed.	Professional translation from Spanish to Basque of the WMT19 ES-EN shared task data	200
Terminology (ICD-10)	EN-EU	Biomed.	Subset of the shared task in-domain ICD-10 terminology set	368
Paracrawl v5 (PCv5)	EN-ES	Open	Subset of the Paracrawl v5 corpus	5k
Elhuyar TMs (ELH)	EU-ES	Open	Data collected from Elhuyar's internal translation memories	1k
WMT18	EU-ES	Biomed.	WMT18 biomedical task test set	277
WMT19	EU-ES	Biomed.	WMT19 biomedical task test set	368

Table 2: Description of the test sets used to evaluate the models.

			Open	In-domain		
System	**Train data**	**Dev data**	**Syn**	**ICD-10**	**EHU**	**PRO**
Baseline1	-	-	15.05	10.02	15.58	13.31
Baseline2	-	-	15.06	10.03	**16.62**	**13.37**
Baseline3	ELH Syn	ELH Syn	**15.28**	9.28	15.08	11.80
Baseline4	ICD-10	ICD-10	0.00	**89.18**	0.00	0.0
SystemA1	ELH Syn (250k); ICD-10	ICD-10	9.33	**90.46**	9.86	6.29
SystemA2	ELH Syn (250k); ICD-10; EHU books	ICD-10; EHU books	10.42	90.26	32.20	8.19
SystemA3	ELH Syn (250k); ICD-10; EHU books	ELH Syn; ICD-10; EHU books	12.72	87.60	13.76	9.38
SystemA4	ELH Syn (all); ICD-10; EHU books	ELH Syn; ICD-10; EHU books	15.47	80.36	24.43	12.95
SystemB1	ELH Syn (all); ICD-10; EHU books; EHU Syn	ELH Syn; ICD-10; EHU books	15.81	82.05	26.45	12.85
SystemB2	ELH Syn (all); ICD-10; EHU books, EHU Syn ; Monoligual	ELH Syn; ICD-10; EHU books	15.69	81.01	26.51	**13.61**
SystemC	ELH Syn (all); ICD-10; EHU books; EHU Syn; Monoligual	ELH Syn; ICD-10; EHU books; Medline Syn 19	**15.91**	83.79	**27.73**	13.50

Table 3: BLEU scores for the English to Basque experiments on out of domain and in-domain test sets.

3.3.2 Results

Table 3 shows BLEU scores for the English to Basque experiments on out of domain and in-domain test sets (Table 2).

As for abstract translation, SystemA4, SystemB1, SystemB2 and SystemC showed a significant improvement on those test sets when compared to the other trained systems. In particular, SystemB2 obtained the best results on PRO test and SystemC the second best result.

The gap between SystemA4 and the other three SystemA's showed that using all the available out of domain data helps avoiding the "catastrophic forgetting" phenomena, where all previous knowledge fades when learning new in-domain examples.

SystemB1 introduces in-domain synthetic data in the training process, which significantly improves the results on the EHU test. This is due to the fact that synthetic data and the EHU test set share the same domain (EHU biomedical books). However, the drop on the PRO test is almost insignificant and it also improves all the baselines on the Synthetic test.

SystemB2 further improves the results on the PRO test set by adding monolingual corpora to the training data and SystemC shows the effect of the validation set by adding more biomedical domain data to the validation set. Both systems improve all the baselines on every test set.

SystemB2 (run1), SystemC (run2) and Baseline2 (run3) were submitted as the best runs for the English to Basque abstract translation task.

Terminology translation task greatly differs from the abstract domain which can be clearly seen in the results. In this case, the best results are obtained by SystemA1 which only includes a small part of the available out of domain data. Furthermore, results are not distant from Baseline4 which was trained with ICD-10 training data. This behaviour shows the specificity of the task where previous complete sentence translation knowledge is not essential.

SystemA2 (run1), SystemA1 (run2) and Baseline4 (run3) were submitted as the best runs for the English to Basque terminology translation task.

3.4 EN-ES experiments

Below we present the different systems trained for the English-Spanish pair and the results obtained

for each of them.

3.4.1 Systems

For the English-Spanish pair we trained the following systems:

Baseline1. We have considered a strong baseline by using Elhuyar's best out of domain EN-ES model. This model was trained on the PCv5 corpus.

SystemA. This model has been trained from scratch mixing in-domain and out domain data. In-domain data comprises previous years' shared tasks Medline data and a subset of the SCIELO dataset. Out of domain data was obtained from the PCv5 corpus.

SystemA'. A variant of SystemA by averaging the three best performing checkpoints.

3.4.2 Results

Table 4 shows BLEU scores for the English to Spanish experiments on out of domain and in-domain test sets (Table 2).

SystemA improves the baseline on all the test sets and SystemA' further improves those results obtaining the best results for the task. Similar to the English to Basque abstract translation task, for the English to Spanish pair adding in-domain data and fine-tuning a previously trained out of domain system improved the results. Furthermore, averaging the first three best checkpoints helped improving the results of the best checkpoint.

SystemA' (run1), SystemA (run2) and Baseline1 (run3) were submitted as the best runs for the English to Spanish abstract translation task.

4 Official results

For the English to Basque abstract translation task we selected the PRO test as the most representative one when choosing the best runs for submission. We assumed that this test set would be the closest to the official task test. The EHU test set was also a great indicator of how robust our system was for the biomedical domain.

BLEU scores were calculated using the multi-eval tool and tokenization as provided in Moses. Table 5 shows the performance of all the submitted runs for the official abstract translation test set. Our submitted run2 has obtained the best score on the official task test, achieving a 0.1279 BLEU score. When compared to other teams, all our submitted runs significantly outperform all the runs. It is worth mentioning that our Baseline2 (run3) which is based on pivoting between two systems (EN-ES and ES-EU) has obtained really close results which indicates that some biomedical knowledge was present in the Spanish to Basque system.

In the case of the terminology task, ICD-10 test set is the official validation task and therefore the most representative one for choosing the best runs.

For the evaluation of terminology we provide two metrics: (i) accuracy, by relying on strict matches (case insensitive) between ground truth ans predictions; and (ii) BLEU score, as measured by the NLTK module sentence_bleu. Table 6 shows the performance of all the submitted runs for the official terminology test set.

Our best submitted run has obtained the best score on the official task test, achieving 0.78 accuracy and a 0.7373 BLEU score. When compared to other teams, all our submitted runs outperform all the runs, except for DCU_MT's run2. It is worth mentioning that out Baseline4 (run3) which was trained on task's English-Basque data has outperformed almost all of the others teams' results. This highlights the improvements obtained by our systems over the baseline.

Finally, for the English to Spanish abstract translation task WMT19 and WMT18 test sets were selected as reference for selecting the best runs.

Table 7 shows the performance of all the submitted runs for the official English to Spanish abstract translation test set. Our best submitted run has obtained the fifth best score on all sentences achieving a 0.4364 BLEU score and the second best team with OK sentences (BLEU=0.4498). In this case, our submitted baseline (run3) also shows a great robustness, indicating some prior biomedical domain knowledge.

5 Conclusions

For the Biomedical Translation Task 2020, we considered several strategies combining open domain and in-domain biomedical data. We have successfully applied transfer learning by fine-tuning a previously available open domain system with in-domain specific data. To tackle the scarcity of English-Basque domain data, we have performed data augmentation by back translating real data.

The systems submitted for the terminology and abstract translation tasks for the English-Basque pair have ranked first on the official task test, achieving 0.78 accuracy for terminology translation and a BLEU of 0.1279 for the translation of

System	Train data	Dev data	Open		In-domain	
			PCv5	ELH	wmt19	wmt18
Baseline1	PCv5	PCv5	**44.69**	35.68	44.27	30.73
SystemA	PCv5 and Biomedical Mix	PCv5 and Biomedical Mix	46.37	36.25	45.72	30.99
SystemA'	PCv5 and Biomedical Mix	PCv5 and Biomedical Mix	**46.54**	**36.29**	**45.74**	**31.22**

Table 4: BLEU scores for the English to Spanish experiments on out of domain and in-domain test sets.

Team	Runs	BLEU
Elhuyar_NLP	run1	0.1271
Elhuyar_NLP	run2	**(1) 0.1279**
Elhuyar_NLP	run3	0.1268
DCU_MT	run1	0.0867
DCU_MT	run2	0.0825
DCU_MT	run3	0.0808
UTS_NLP	run1	0.0530
UTS_NLP	run2	0.0549
UTS_NLP	run3	0.0528
Ixamed	run1	0.0815
Ixamed	run2	0.0782
Ixamed	run3	0.0884
Baseline	-	0.0596

Table 5: Performance scores on the official English to Basque abstract translation test set.

Team	Runs	Accuracy	BLEU
Elhuyar_NLP	run1	**(1) 0.78**	**(1) 0.7373**
Elhuyar_NLP	run2	0.77	0.7356
Elhuyar_NLP	run3	0.75	0.7229
DCU_MT	run1	0.73	0.7083
DCU_MT	run2	0.76	0.7239
DCU_MT	run3	0.75	0.7179
UTS_NLP	run1	0.73	0.7115
UTS_NLP	run2	0.73	0.7122
UTS_NLP	run3	0.73	0.7085
Ixamed	run1	0.12	0.1314
Ixamed	run2	0.08	0.0721
Ixamed	run3	0.13	0.1481

Table 6: Performance scores on the official terminology test set.

abstracts. For the English to Spanish abstract translation task our systems have obtained competitive enough results, being the second team for OK sentences (BLEU=0.4498).

In all cases, even developed baselines have achieved outstanding results. For the English-Spanish task, Paracrawl v5 has proven to be a robust baseline for biomedical domain systems, as it seems to contain some biomedical crawled websites. For the English-Basque task, fine-tuning one

Team	Runs	BLEU	BLEU OK
Elhuyar_NLP	run1	(5) 0.4364	(4) 0.4498
Elhuyar_NLP	run2	0.4359	0.4493
Elhuyar_NLP	run3	0.4263	0.4394
Ixamed	run1	0.4052	0.4171
Ixamed	run2	0.3729	0.3836
Ixamed	run3	0.3755	0.3858
Sheffield	run1	**0.4493**	0.4493
TRAMECAT	run1	0.4238	0.4361
UNICAM	run1	0.4434	0.4572
UNICAM	run2	0.4464	**0.4672**
UNICAM	run3	0.4453	0.4662
Baseline	-	0.3709	0.3813

Table 7: Performance scores on the official English to Spanish test set.

of the pivoting pairs (EN-ES) we have created a robust baseline for the biomedical domain.

Adding monolingual corpora to the training data, as copied target, seems to improve the decoder by adapting the systems to better perform on domain specific terminology. Even though some noise is introduced by copying the target to the source side, the results are improved.

Terminology task showed promising results when translating biomedical domain terms, which could lead to a production ready system.

References

Chenhui Chu, Raj Dabre, and Sadao Kurohashi. 2017. An empirical comparison of domain adaptation methods for neural machine translation. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 385–391.

Anna Currey, Antonio Valerio Miceli-Barone, and Kenneth Heafield. 2017. Copied monolingual data improves low-resource neural machine translation. In Proceedings of the Second Conference on Machine Translation, pages 148–156.

Miquel Esplà, Mikel Forcada, Gema Ramírez-Sánchez, and Hieu Hoang. 2019. ParaCrawl: Web-scale parallel corpora for the languages of the EU. In Proceedings of Machine Translation Summit XVII

Volume 2: Translator, Project and User Tracks, pages 118–119, Dublin, Ireland. European Association for Machine Translation.

Abdul Khan, Subhadarshi Panda, Jia Xu, and Lampros Flokas. 2018. Hunter nmt system for wmt18 biomedical translation task: Transfer learning in neural machine translation. In Proceedings of the Third Conference on Machine Translation: Shared Task Papers, pages 655–661.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M Rush. 2017. Opennmt: Open-source toolkit for neural machine translation. arXiv preprint arXiv:1701.02810.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. arXiv preprint arXiv:1706.03872.

Jaehong Park, Jongyoon Song, and Sungroh Yoon. 2017. Building a neural machine translation system using only synthetic parallel data. arXiv preprint arXiv:1704.00253.

Hassan Sajjad, Nadir Durrani, Fahim Dalvi, Yonatan Belinkov, and Stephan Vogel. 2017. Neural machine translation training in a multi-domain scenario. arXiv preprint arXiv:1708.08712.

Danielle Saunders, Felix Stahlberg, and Bill Byrne. 2019. Ucam biomedical translation at wmt19: Transfer learning multi-domain ensembles. arXiv preprint arXiv:1906.05786.

Rico Sennrich, Alexandra Birch, Anna Currey, Ulrich Germann, Barry Haddow, Kenneth Heafield, Antonio Valerio Miceli Barone, and Philip Williams. 2017. The university of edinburgh's neural mt systems for wmt17. arXiv preprint arXiv:1708.00726.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Improving neural machine translation models with monolingual data. arXiv preprint arXiv:1511.06709.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Xabier Soto, Olatz Perez-de Viñaspre, Maite Oronoz, and Gorka Labaka. 2019. Leveraging snomed ct terms and relations for machine translation of clinical texts from basque to spanish. In Proceedings of the Second Workshop on Multilingualism at the Intersection of Knowledge Bases and Machine Translation, pages 8–18.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In Advances in neural information processing systems, pages 5998–6008.

Olatz Perez-de Viñaspre. 2017. Automatic medical term generation for a low-resource language: translation of SNOMED CT into Basque. Ph.D. thesis, PhD thesis, University of the Basque Country, Donostia, Euskal Herria.

Rui Wang, Andrew Finch, Masao Utiyama, and Eiichiro Sumita. 2017. Sentence embedding for neural machine translation domain adaptation. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 560–566.

Antonio Jimeno Yepes, Aurélie Névéol, Mariana Neves, Karin Verspoor, Ondřej Bojar, Arthur Boyer, Cristian Grozea, Barry Haddow, Madeleine Kittner, Yvonne Lichtblau, et al. 2017. Findings of the wmt 2017 biomedical translation shared task. In Proceedings of the Second Conference on Machine Translation, pages 234–247.

Jiajun Zhang and Chengqing Zong. 2016. Exploiting source-side monolingual data in neural machine translation. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 1535–1545.

YerevaNN's Systems for WMT20 Biomedical Translation Task: The Effect of Fixing Misaligned Sentence Pairs

Karen Hambardzumyan[1], Hovhannes Tamoyan[1,2], and Hrant Khachatrian[1,3]

[1]YerevaNN
[2]American University of Armenia
[3]Department of Informatics and Applied Mathematics, Yerevan State University

Abstract

This report describes YerevaNN's neural machine translation systems and data processing pipelines developed for WMT20 biomedical translation task. We provide systems for English-Russian and English-German language pairs. For the English-Russian pair, our submissions achieve the best BLEU scores, with en→ru direction outperforming the other systems by a significant margin. We explain most of the improvements by our heavy data preprocessing pipeline which attempts to fix poorly aligned sentences in the parallel data.

1 Introduction

Biomedical machine translation is a perfect playground to develop narrow domain neural machine translation models. In such tasks, the available parallel in-domain data is usually limited and noisy which creates many challenges.

In the previous works (Bawden et al., 2019), researchers focused on transfer learning methods (Saunders et al., 2019) or attempted to mix the training data with other sources (Peng et al., 2019) to address the issue of data scarcity. In this work, we show that the transfer performance is very dependent on the quality of the training data, and with a little effort, it is possible to improve the given MEDLINE training data and gain a significant performance boost.

We have manually created a much higher quality subset of the original MEDLINE training data for local evaluation purposes. The insights collected during this manual analysis was then used to fix the most common issues within the training data. In particular, we noticed that the original dataset contained paper abstracts in two languages without sentence-level alignments, and the training corpus provided by the organizers was created using an automated sentence segmentation and alignment process, which was not perfect. We built a data pipeline[1] that handles 1) cleanup, 2) sentence segmentation, 3) alignment of translation sentence pairs and 4) preprocessing.

In our experiments, we did not use any data source other than MEDLINE. We chose our baseline model and two other models with the highest BLEU scores on a local test set as our three submissions. The best ones got 35.2% BLEU on English-German and 41.3% on German-English test sets. For English-Russian and Russian-English directions we reached BLEU scores of 37.9% and 43.2% respectively, which are the best scores among all submissions of WMT20 Biomedical Translation Task. Moreover, our models are cheap to train: the average training time of our best models is approximately 30 minutes on a single NVIDIA Titan V GPU.

The paper is organized as follows: Section 2 presents fine-tuning details and evaluation methods for our NMT systems, Section 3 describes the data used in the experiments and the data processing pipeline, Section 4 presents our novel method of monotonic alignment based on multilingual language models. Section 5 discusses the results.

2 System Description

2.1 Pretrained Models

All our NMT models are built on top of WMT19 News Translation task winner models by Ng et al.. We employ FairSeq library (Ott et al., 2019) to fine-tune pretrained models on the in-domain translation data.

The pretrained models are based on `transformer_wmt_en_de_big` architecture (Vaswani et al., 2017) with a modified feedforward

[1]Our data pipeline is available at https://github.com/YerevaNN/parasite

Proceedings of the 5th Conference on Machine Translation (WMT), pages 820–825
Online, November 19–20, 2020.

dimension (8192) and a shared matrix for input and output embeddings. Additionally, en↔de models share vocabulary and embeddings for both source and target sides.

2.2 Fine-Tuning

We start fine-tuning `single_model` versions of Facebook's WMT19 models[2] on in-domain parallel data and stop the training when the perplexity on the validation set does not improve for 5 consecutive epochs.

To fight noisy training data we use label-smoothed cross-entropy loss (Müller et al., 2019).

The neural architecture and related implementation details cannot be changed in the fine-tuning scenario. While this limited our experimental setup, however, it also allowed us to care less about hyperparameter tuning and focus on other parts of the pipeline.

2.3 Implementation Details

The hyperparameters for our baseline models (`run1`) are as follows. The models are fine-tuned on the training data using an inverse-square-root learning rate schedule with 4000 warm-up steps with an initial learning rate of 10^{-5}. Instead of using a fixed batch size, we make batches of maximum 3584 tokens to fit in the memory. For label smoothing, we set a smoothing coefficient of 0.1. Unlike the pretrained models, we use standard Adam betas and disable dropout.

Training with bigger batches (implemented using gradient accumulation, a single update per 128 batches) not only helped us to reduce total training time 4x but also resulted in better models (including our best submissions `run2` and `run3`).

All the models are trained on a single NVIDIA Titan V GPU with 16-bit floating-point operations. The average duration of fine-tuning with bigger batches was 30 minutes.

Finally, we use a beam size of 32 in the inference mode.

2.4 Evaluation and Model Selection

We use two kinds of validation sets for model selection. For early stopping, we calculate the perplexity on a regular validation set which is extracted from the training data. To determine our best models for submissions, we use a separate in-domain dataset which we call "local test set" and calculate BLEU score on it. All BLEU scores are calculated with SacreBLEU case-insensitive configuration.

3 Data

3.1 Parallel Data

For all directions, we use only MEDLINE training data provided by the shared task organizers. We take random 50 documents from the training data as the validation set. In case of en↔de we use OK-tagged sentence-pairs from WMT'19 biomedical translation test set (Bawden et al., 2019) as the *local test set*. To have a *local test set* of a similar quality for en↔ru, we take another random 50 documents, then manually fix misaligned sentences and filter out a few pairs with incorrect translations.

During the manual review of the en↔ru *local test set* we noticed that the provided data was poorly aligned, and it was possible to get high-quality sentence pairs by re-aligning the sentences (only 9 sentences were dropped except the titles/subtitles, out of 504 sentences). Then we tried to use these insights to build a new automated system for monotonic alignment of the sentences (described in Section 4).

Table 1 exhibits the most common issues found in the MEDLINE training data:

- The bitext documents may be misaligned: the translation of a source sentence may appear on a different line, or even on multiple lines, in the target side,
- Headings and section names may occur next to a sentence on one side only, or on both sides,
- English documents may start with titles (often wrapped in brackets), while the Russian ones do not.

These issues are too common in the training set, and simply removing incorrect pairs of sentences would significantly reduce the dataset. Instead, we decided to fix the misaligned sentences to preserve as much parallel content as possible. The solution is described in Section 4.

3.2 Monolingual Data

Although the base models we use are already trained with backtranslation, we try to fine-tune with backtranslation as well. We obtain translations

[2] `wmt19.en-de.joined-dict.single_model` `wmt19.de-en.joined-dict.single_model`, `wmt19.en-ru.single_model`, `wmt19.ru-en.single_model`

1	*[Risk factors of stroke in men exposed to environmental factors at workplace].* OBJECTIVE	Цель исследования - изучение факторов риска развития инсульта у мужчин разных возрастных групп, подвергающихся воздействию неблагоприятных производственных факторов.
2	To explore risk factors of stroke in men of different age groups exposed to adverse environmental factors at work.	*Материал и методы.*
3	*MATERIAL AND METHODS* Four hundred and eleven men after stroke, aged from 30 to 65 years, including 335 patients, who had been exposed to adverse environmental factors at work, were compared to 76 patients who had not been exposed to adverse environmental factors.	Обследованы 411 мужчин в возрасте от 30 до 65 лет, перенесших инсульт, из них 335 пациентов подвергались влиянию неблагоприятных производственных факторов и 76 пациентов, которые воздействия вредных факторов не испытывали (группа сравнения).
4	*RESULTS*	*Результаты.*
5	The distribution of the frequencies of risk factors of stroke depending on the character of adverse factors was shown.	Установлена частота распределения факторов риска развития инсульта у мужчин в зависимости от характера профессиональных вредностей.

Table 1: A hand picked example from MEDLINE en↔ru training set (document #26978637) which demonstrates the most common issues in the dataset. The first line in English includes the title of the paper which is not present in Russian. The English version of the main content of the first line in Russian is given on the second line. Line 3 in English has an extra heading which corresponds to Line 2 on the right side. The rest of the third line on the left matches to the third line on the right side, and the last two lines are correct.

with the fine-tuned models mentioned above, then fine-tune *new* models on a mixed data consisting of the regular parallel training data and backtranslated data with equal proportions.

To perform backtranslation we need a set of in-domain monolingual sentences that do not overlap with the test set. To train backtranslated de↔en and ru ↔ en directions, we took all English sentences from all parallel corpora available from MEDLINE (both training and test sets) excluding the parallel corpora we would eventually train on. This way we collected 296,052 (236,379) English sentences for German (for Russian). To obtain a parallel corpus we translated them using our models, and then filtered them using the same process as with the regular training data (see the next subsection). We ended up with 281,054 (220,916) sentence pairs for en↔de (en↔ru).

We did not perform backtranslation from Russian or German (directions en→de and en→ru), as we did not expect to find in-domain sentences that are not present in MEDLINE.

When translating the monolingual sentences, we tried `sampling`, `sampling-top5`, `greedy`, `beam`, `beam+noise` decoding methods similar to (Edunov et al., 2018), but no major difference in terms of the final BLEU score has been observed.

3.3 Preprocessing

The preprocessing pipeline for our models has to be identical to the one used for pretrained models. First, we perform punctuation normalization (quotation, commas, numbers, replacing punctuation and removing control characters) using `SacreMoses` library. Then, we tokenize the resulting sentences using Moses (Koehn et al., 2007) tokenizer with aggressive dash splits and escaping XML entities. Finally, we use subword segmentation (Sennrich et al., 2016) (`fastbpe` implementation) with BPE codes from pretrained models, with 24k and 32k splits for Russian and for joint English & German, respectively.

We perform additional filtering of the parallel data before the training: we skip those sentence pairs where 1) source or target sentence has more than 250 subwords and/or 2) the ratio of lengths of the source and target sentences is more than 3/2.

During inference, we truncate sentences to the first 1024 subwords (the number of the positional embeddings).

During our early experiments we noticed several issues with our preprocessing pipeline which we fixed for the later experiments. In particular, we noticed that some `sacremoses` command line flags were broken, and the out-of-the-box inference tool from `FairSeq` did not fully replicate the preprocessing pipeline used for training (punctuation normalization and vocabulary-aware subword segmentation). The original pipeline (called *v1*) was used for our baseline models. The later experiments used the fixed implementations of `sacremoses` and `FairSeq` (denoted by *v2*).

4 Monotonic Alignments

The problems of the training set described in Section 3.1 can be caused by poor 1) XML parsing, 2) sentence segmentation, or 3) monotonic segment alignment method. Here we describe a novel method for monotonic sentence alignment using multilingual language models and discuss the contribution of its hyperparameter choices. Multilingual language models have been previously shown to be effective in parallel data mining (Kvapilíková et al., 2020). We also compare our approach to the baseline data pipeline by the shared task organizers which is based on `Syntok` segmentation system and `GMA` (Melamed, 2001).

Our method of monotonic sentence alignments is as follows: we calculate a similarity matrix of all source-target candidate pairs and decode pairs to maximize the similarity of the resulting sentence pairs. We consider two approaches for the decoding step: greedy and dynamic.

4.1 Similarity Matrix

The similarity matrix is calculated using Euclidean distances of sentence embeddings from a pretrained multilingual language model. We found `xlm-roberta-large` (Conneau et al., 2019) to be the best one. In order to obtain a fixed size vector for each sentence, we simply take the average of the wordpiece embeddings (Cer et al., 2018; Artetxe and Schwenk, 2019).

We also attempt to address some common issues concerning the given MEDLINE abstracts that may harm the quality of the alignments: 1) we remove titles from the English version that are absent in the Russian version, 2) we detect the headings that often get attached to adjacent sentences, 3) we lowercase the text before obtaining embeddings (as the English headings are written in capitals, unlike the Russian ones), 4) we experiment with different sentence segmentation systems such as SciSpacy (Neumann et al., 2019) (in-domain, for English) and Razdel[3] (focused on Russian), 5) we also penalize candidates with source/target length ratios exceeding 2. Additionally, we consider using normalized distances and the margin based approach described in (Artetxe and Schwenk, 2019).

[3] `https://github.com/natasha/razdel`

4.2 Greedy Approach

In greedy approach, we construct the set of correct sentence pairs in an iterative process. Given the similarity matrix, at each step we add the sentence pair with the maximum similarity score. As there is an assumption that the alignments should be monotonic, after each step we exclude all remaining candidate sentence pairs that would break the monotonicity. Our implementation finds at most one target sentence for a source sentence (and vice versa).

Algorithm 1: Greedy decoding

$S_N, T_M \leftarrow$ source and target sentences
$D_{i,j} \leftarrow Sim(S_i, T_j)$
$Res \leftarrow \{\}$
while $|Align| < \min(N, M)$ **do**
  $i, j \leftarrow \arg\max(D)$
  $Align \leftarrow Align \cup \{i, j\}$
  $D_{i..N,0..j}, D_{0..i,j..M} \leftarrow 0$
end
Result: Align

4.3 Dynamic Algorithm

In the dynamic algorithm, we consider maximizing the sum of the similarity scores of the selected sentence pairs according to the given matrix. Our implementation of this approach, unlike the greedy one, can produce sentences consisting of multiple (up to K) segments on each side. To find the mapping with the best total similarity score we use dynamic programming.

5 Results

For WMT20 Biomedical Translation Task we prepared three submissions: `run1` for all directions was the baseline model, while for `run2` and `run3` we chose the best models according to their BLEU score on the local test set at the time of the submission. In `run2` and `run3`, all the models besides de→en of `run2` are trained with our data pipeline and bigger batches. The official BLEU scores on samples with "OK" aligned sentences alongside with our local test set are presented in Table 2.

For de→en of `run2`, backtranslation data was collected with beam search (size of 8), in case of ru→en, we had noise added similar to Edunov et al., and for `run3` we used a simple sampling strategy. Our experiments with backtranslation showed no

BLEU Scores on WMT20 Test / Local Test				
Models	**en→de**	**de→en**	**en→ru**	**ru→en**
run1	35.2 / 34.5	41.3 / 45.4	32.6 / 27.7	NA / 30.7
run2		41.4 / 44.7	39.4 / 31.6	43.3 / 33.0
run3	35.2 / 35.1	41.3 / 45.6	37.9 / 31.8	43.2 / 33.1

Table 2: BLEU scores of our submissions

Algorithm 2: One-to-many (K) dynamic decoding

$S_N, T_M \leftarrow$ source and target sentences
$Best_{N,M} \leftarrow 0$
$Res_{N,M} \leftarrow \{\}$
for $i = 1 \rightarrow N$ **do**
  for $j = 1 \rightarrow M$ **do**
    for $u, v = 1 \rightarrow K$ **do**
      $candidate \leftarrow Best_{i-u,j-v} + Sim(S_{i-u..i}, T_{j-v..j})$
      if $candidate > Best_{i,j}$ **then**
        $Best_{i,j} \leftarrow candidate$
        $Res_{i,j} \leftarrow Res_{i-k,j} \cup \{S_{i-k..i}, T_{j-v..j}\}$
      end
    end
  end
end

significant advantage of any of those compared to the others.

For `run2` and `run3`, we used v2 preprocessing, the sentence splitting was done with `scispacy` (for English and German) and a slightly modified version of `razdel` (for Russian).

After our submissions, we further improved our data pipeline. Table 3 is an empirical analysis of the effect of different components of our data pipeline, as measured by the performance on the final translation task. Each row of the table corresponds to a model trained on the data obtained from a pipeline with certain components enabled. There is no other between the rows, all models are trained by fine-tuning the general domain baseline using our default hyperparameters. We measure the BLEU score on the local test set.

Fixing the issues of the standard preprocessing (v2 vs. v1) gives a significant boost, especially when decoding to Russian (en→ru direction). The effect of training with bigger batch sizes gives only a slight improvement, while the absolute training duration reduces drastically.

Model	**en→ru**	**ru→en**
baseline model	27.7	30.7
+ v2 preprocessing	30.5	31.3
+ train with bigger batches	30.7	31.3
+ greedy alignments	30.1	31.8
+ detect section names	30.7	32.3
+ remove titles	31.3	**32.5**
+ optimize total similarity	30.4	32.2
+ normalize distance matrix	30.8	32.1
+ penalize source/target ratio	31.2	31.5
+ one-to-many (K=3)	**32.2**	32.3

Table 3: The effect of different components of the data processing pipeline. We report BLEU scores on the local test set.

As mentioned previously, there were issues with section names and titles in the provided parsed documents. After addressing these issues, our greedy approach gives better alignments.

The total similarity optimization using dynamic programming is not always better than the greedy method, but the performance improves for en→ru with another +1.1% BLEU score. Overall, the new data pipeline gives an enhancement in NMT performance: +1.6% BLEU for ru→en and a bigger gain of +4.5% BLEU score for en→ru.

Although we observe consistent performance improvement for both directions en↔ru, the effect for en→ru direction is more significant. We could not determine the reason for such assymmetry.

6 Conclusion

This work presents the systems our team developed for English-German and English-Russian language pair tracks of WMT20 Biomedical Translation Task. We achieve the best results on the official test set for English↔Russian language pair, outperforming competitors by a significant margin on English→Russian direction. We show that it is possible to improve the performance of neural machine translation models by simply improving the quality of the in-domain parallel

data. The suggested method for monotonic sentence segment alignment based on pretrained multilingual language models demonstrated promising results. We explored how different components of our data processing pipeline contributed to the quality of the resulting translation systems. In future work, we plan to investigate the applicability of this pipeline to a wider set of language pairs and domains.

Acknowledgments

We would like to thank Adam Bittlingmayer from ModelFront for useful discussions on the quality of parallel corpora. All experiments were performed on Titan V GPUs donated to YerevaNN by NVIDIA.

References

Mikel Artetxe and Holger Schwenk. 2019. Margin-based parallel corpus mining with multilingual sentence embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3197–3203.

Rachel Bawden, Kevin Bretonnel Cohen, Cristian Grozea, Antonio Jimeno Yepes, Madeleine Kittner, Martin Krallinger, Nancy Mah, Aurelie Neveol, Mariana Neves, Felipe Soares, et al. 2019. Findings of the wmt 2019 biomedical translation shared task: Evaluation for medline abstracts and biomedical terminologies. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 29–53.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. 2018. Universal sentence encoder for english. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 169–174.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. *arXiv preprint arXiv:1808.09381*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*, pages 177–180. Association for Computational Linguistics.

Ivana Kvapilíková, Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Ondřej Bojar. 2020. Unsupervised multilingual sentence embeddings for parallel corpus mining. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 255–262.

I Dan Melamed. 2001. Geometric approach to mapping bitext correspondence.

Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. 2019. When does label smoothing help? In *Advances in Neural Information Processing Systems*, pages 4694–4703.

Mark Neumann, Daniel King, Iz Beltagy, and Waleed Ammar. 2019. Scispacy: Fast and robust models for biomedical natural language processing. *arXiv preprint arXiv:1902.07669*.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook fair's wmt19 news translation task submission. *arXiv preprint arXiv:1907.06616*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. *arXiv preprint arXiv:1904.01038*.

Wei Peng, Jianfeng Liu, Liangyou Li, and Qun Liu. 2019. Huawei's nmt systems for the wmt 2019 biomedical translation task. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 164–168.

Danielle Saunders, Felix Stahlberg, and Bill Byrne. 2019. Ucam biomedical translation at wmt19: Transfer learning multi-domain ensembles. *arXiv preprint arXiv:1906.05786*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Pretrained Language Models and Backtranslation for English-Basque Biomedical Neural Machine Translation

Inigo Jauregi Unanue
University of Technology Sydney
RoZetta Technology
inigo.jauregi
@rozettatechnology.com

Massimo Piccardi
University of Technology Sydney
massimo.piccardi@uts.edu.au

Abstract

This paper describes the machine translation systems proposed by the University of Technology Sydney Natural Language Processing (UTS_NLP) team for the WMT20 English-Basque biomedical translation tasks. Due to the limited parallel corpora available, we have opted to train a BERT-fused NMT model that leverages the use of pretrained language models. Furthermore, we have augmented the training corpus by backtranslating monolingual data. Our experiments show that NMT models in low-resource scenarios can benefit from combining these two training techniques, with improvements of up to 6.16 BLEU percentage points in the case of biomedical abstract translations.

1 Introduction

Nowadays, most of the literature and scientific terminology produced in the biomedical field is in English, which limits the access to this information by non-English speaking researchers, doctors and patients. Thus, it would be very useful to avail of machine translation systems that can effectively translate this information into other languages, so that more people can be able to access it and benefit from it.

However, many of the world languages lack sufficient parallel corpora to properly train machine translation systems in this domain. State-of-the-art neural machine translation (NMT) models (Sutskever et al., 2014; Bahdanau et al., 2015) suffer from overfitting when trained on insufficient data, and thus fail to generate accurate translations (Koehn and Knowles, 2017).

In this paper we address this problem for a low-resource language, Basque. We have taken part in the WMT20 Biomedical Translation challenge, which has released two interesting shared tasks involving this language, namely, the English-to-Basque translation of biomedical article abstracts and the English-to-Basque translation of medical terminology. In order to overcome the issue of having limited supervised training data, we have decided to apply two promising ideas proposed in the literature. First, we have applied transfer learning by training a *BERT-fused* NMT model (Zhu et al., 2020) that uses source-language contextual embeddings inferred by a pretrained language model (LM) as additional input features, both in the encoder and in the decoder. Second, we have augmented the training corpus using backtranslation (Sennrich et al., 2016; Burlot and Yvon, 2018). For this, a BERT-fused NMT model has been trained in the opposite translation direction (Basque → English) to translate sentences from large monolingual corpora (e.g. Wikipedia, medical texts).

The experiments have shown that an NMT baseline can greatly benefit from combining these training techniques. The three best performing systems in both tasks (terminology and abstracts) have been submitted to the WMT20 biomedical translation shared task under the UTS_NLP team name.

2 Related Work

2.1 Pretrained LMs

Pretrained LMs have been one of the most remarkable advancements in transfer learning for NLP in recent years. They are large neural networks that are trained over massive datasets (millions of sentences) in an unsupervised manner, and can effectively learn the regularities/patterns of a language. Then, such general networks can be applied to efficiently train smaller networks for downstream tasks, using much smaller annotated datasets. ELMo (Peters et al., 2018), BERT (Devlin et al., 2019) and GPT-2 (Radford et al., 2018) are some examples of pretrained LMs that have

Proceedings of the 5th Conference on Machine Translation (WMT), pages 826–832
Online, November 19–20, 2020.

achieved state-of-the-art results in various natural language understanding tasks such as, among others, sentiment analysis, paraphrase detection, and question answering.

In NMT, various recent works have proposed incorporating pretrained LMs into the standard encoder-decoder architecture. Lample and Conneau (2019) have proposed pretraining an LM with a novel "cross-lingual LM objective" that uses parallel training data to predict masked words of either language. Edunov et al. (2019) have replaced standard word embeddings with contextual word embeddings learned by a pretrained LM. Their experiments have showed that contextual embeddings are more effective in the encoder and when fixed ("ELMo-style augmentation"). Conversely, Clinchant et al. (2019) have initialized the encoder of an NMT model with the weights of a pretrained LM network, and fine-tuned them ("GPT-2/BERT-style augmentation"), reaching similar performance, but faster convergence. Yang et al. (2020) have added the contextual embeddings of a pretrained LM as additional input features to the standard embeddings, and incorporated a dynamic switch to let the model learn how to weigh each input. Finally, the BERT-fused NMT model (Zhu et al., 2020) follows a similar idea, by adapting the architecture of the transformer network in order to have an extra self-attention layer that learns to weigh the contextual embeddings of the LM. This attention-layer is seamlessly added to both the encoder and the decoder. Given the improvement in performance achieved by the BERT-fused NMT on several datasets and the well-supported code by the authors (built on top of *fairseq*), we have decided to adopt this model in our experiments.

2.2 Backtranslation

In NMT, backtranslation (Sennrich et al., 2016; Burlot and Yvon, 2018) has become a common approach to alleviate the problem of having limited parallel data for training. It consists of first training a *target* $\rightarrow$ *source* NMT model with the available parallel corpus. Then, this model is used to translate a large number of sentences from monolingual corpora in the target language, which are usually more available than parallel corpora, to the source language. The resulting "silver corpus" is used as additional training data for the *source* $\rightarrow$ *target* NMT model, and can often help to boost the fluency of the generated translations.

However, the most effective way of using backtranslation is still an open research question. Poncelas et al. (2018) have explored different combinations of backtranslated and human-translated datasets, and have found that in their low-resource scenario a 2:1 backtranslated-to-human-translated sentences ratio is optimal; beyond that, increasing the size of the backtranslated data deteriorates the performance. Edunov et al. (2018) have shown that backtranslating by either sampling from the model or adding noise to a standard beam search can improve the final translation accuracy substantially. Burlot and Yvon (2018) have generated more natural *pseudo-source* sentences by training a generative adversarial network (GAN). Finally, Soto et al. (2020) have found that combining backtranslated data from different sources (i.e. out-of-domain data, in-domain data) and different models (i.e. rule-based, SMT, NMT) can also improve the accuracy of the final translations. In our work, we have explored using a BERT-fused NMT model for backtranslating monolingual data, expecting that the transfer learning achieved from using pretrained LM in Basque will produce better quality pseudo-source sentences.

3 Resources

All the experiments have been carried out using only the parallel and monolingual training data recommended by the organisers on the shared task website. Table 1 summarizes all the data used in our experiments.

3.1 Parallel Data

The medical terminology translation task consists on translating ICD-10 (International Clasification of Diseases) code descriptions from English to Basque. The descriptions are relatively short sentences (8 tokens on average). The organizers have provided an in-domain parallel corpus for training and validation. The blind test set contains 2,000 sentences in English.

The abstract translation task involves translating sentences of abstracts from biomedical scientific research papers. The sentence in this task are longer compared to those in the terminology task (24 tokens on average). However, for this task the organizers have not provided any in-domain parallel data for training or validation, only 375 English sentences for blind testing. Consequently, we have decided to form a small validation set from

	train	dev	test
in-domain (IO)			
ICD-10	25,900	2,000	2,000
abstracts	-	50	375
out-of-domain (OOD)			
EhuHac	550,000	-	-
QED	16,000	-	-
TED talks	5,623	-	-

(a) Parallel data.

general	
wikipedia	1.5M
biomedical	
wikipedia biomedical	8,000
hospital notes	2,000
Snomed CT	50,000

(b) Monolingual data in Basque.

Table 1: Number of sentences in each dataset.

the English-Italian biomedical abstract translation dataset provided on the website. We have selected 50 sentences in English from that dataset and translated them manually into Basque. In this way, we have managed to assemble a small, yet high-quality validation dataset in a domain similar to that of the actual task, and used it for selecting the best models for submission.

Finally, we have used out-of-domain (OOD) parallel corpora to compensate for the lack of in-domain training data. From the data provided by the organizers, we have used: the *EhuHac* dataset, which consists of translations of 136 fiction books; the *QED* dataset, which are translations of subtitles for educational videos and lectures; and the *TED talks* dataset, containing transcripts of TED talk videos.

3.2 Monolingual Data

The monolingual data have been grouped in two categories. First, we have the *general* domain texts, which include 1.5M sentences from the Basque Wikipedia. Second, we have the *biomedical* domain texts, which include a group of medical articles from the Basque Wikipedia and hospital notes written by doctors. We have applied backtranslation to generate pseudo-parallel datasets from these monolingual data. A BERT-fused NMT model has been trained over the available OOD parallel data in the reverse translation direction, and applied to translate Basque sentences to English. For more details on the training of the BERT-fused NMT model, please see Section 4.

Additionally, we have included as part of the biomedical monolingual data the subset of Basque SNOMED CT terms provided by the organizers, which have been automatically translated from English using a rule-based machine translation system. Using the IDs of the terms, we have been able to match them with the original English terms and include them as additional training data.

3.3 Pretrained BERT Models

We have explored using several different pretrained BERT LMs to include them in our BERT-fused NMT model. The pretrained BERT models have been downloaded from Hugging Face[1] :

- **bert-base-uncased** : Original BERT LM model proposed by Devlin et al. (2019). Pretrained on the BookCorpus (800M words) (Zhu et al., 2015) and the English Wikipedia (2,500M words).
- **bert-pubmed**: Pretrained over biomedical articles and journals collected from PubMed. There is no clear description of the amount of data used for pretraining. Hugging Face model name: `monologg/biobert_v1.1_pubmed`.
- **bert-mimic-pubmed**: Pretrained over biomedical articles and journals collected from PubMed and electronic health records of intensive care unit patients from MIMIC-III (Johnson et al., 2016). There is no clear description of the amount of data used for pretraining. Hugging Face model name: `adamlin/NCBI_BERT_pubmed_mimic_uncased_base_transformers`
- **bert-discharge-summaries**: Pretrained model proposed by Alsentzer et al. (2019) trained on all discharge summaries from MIMIC-III. Hugging Face model name: `emilyalsentzer/Bio_Discharge_Summary_BERT`.

Pretrained LM for backtranslation:

- **berteus-base-cased**: Pretrained LM in Basque (Agerri et al., 2020) trained over the Basque Media Corpus (BMC) (224M words). Hugging Face model name: `ixa-ehu/berteus-base-cased`.

[1] https://huggingface.co/models

4 Training and Hyperparameter Tuning

We have trained a BERT-fused NMT model (Zhu et al., 2020) with the open-source code provided by the authors[2], which is built on top of *fairseq*[3]. Following the authors recommendation, as a *warmup* step, first a standard transformer-based (Vaswani et al., 2017) NMT model has been trained over the training data. Then, this model has been used as both a baseline and to initialize the weights that the BERT-fused NMT model has in common. We have used the `transformer_iwslt_de_en` architecture as the NMT model, which consists of a 6-layer transformer network as the encoder and the decoder, with the embedding dimension set to 512 and the hidden layer dimension to 1024. Additionally, we have used the following training hyperparameters: dropout 0.1, label-smoothing 0.1, `inverse_sqrt` learning scheduler, warmup updates $4,000$, warmup initialization learning rate $1e^{-7}$, minimum learning rate $1e^{-9}$, weight decay 0.0001, BERT encoder dropout 0.5 and the Adam optimizer (Kingma and Ba, 2015). The learning rate $[0.0002, 0.00002]$ and the number of tokens per batch $[1024, 4048]$ have been tuned using the validation set. All the datasets have been lowercased and tokenized using the *moses tokenizer*. Additionally, we have learned subword units using Byte Pair Encoding (BPE) (Sennrich et al., 2015) with $10,000$ merge operations in order to reduce the vocabulary size and handle unknown words.

In the terminology translation task we have only used the in-domain ICD-10 code description data to train our models, because adding additional OOD parallel data or backtranslated data was degrading the performance of the model. This is probably likely due to the specific and structured language used in the code descriptions, which is very different from the rest of the available texts. The baseline NMT was warmed up for 50 epochs and the best model over the validation was selected. Then, the BERT-fused models was tuned for 10 more epochs.

In the abstract translation task, due to the fact that in-domain parallel data were not available, we have explored training the model with different combinations of the OOD parallel data, the ICD-10 training data and the backtranslated data. The ICD-10 data and the backtranslated biomedical data have been upsampled x5 and x10, respectively. In this task, the baseline NMT was warmed up for 30 epochs, as the training data are much larger (longer training times) and because we have seen no noticeable improvement after the 30th epoch. Like in the previous task, the BERT-fused models have been tuned for another 10 epochs over the best baseline model.

Evaluation of the models has beeen carried out using the standard BLEU metric (Papineni et al., 2002). In the case of the terminology translation task, we have also used a case-insensitive strict accuracy metric, in which an ICD-10 code description is considered correct only if it is a complete string match with the reference (no partial scores).

5 Results

5.1 Terminology Translation

In the terminology translation task (Table 3a) all the models have achieved high numerical results over the validation set ($> 73\%$ accuracy and > 88.7 BLEU). In terms of translation scores, one could say that this was an easy task and that it is almost solved. However, we would like to argue that this is not the case. Compared to other translation tasks (e.g. abstracts, news, TED talks), the space of correct translations is much smaller in the ICD-10 task since even a single-word mistake (e.g. *abscess of bursa,* ***right*** *shoulder* VS *abscess of bursa,* ***left*** *shoulder*) may result in a misunderstanding with serious consequences. Therefore, there is still margin for improvement.

On the other hand, we have observed that the BERT-fused NMT models have consistently outperformed the baseline, on average by $+1.61$ percentage points (pp) of accuracy and by $+0.7$ pp of BLEU. All pretrained BERT LMs have achieved comparable results, yet surprisingly the *bert-base-uncased* model has proved the best, despite being the only LM that had not been pretrained on biomedical data.

5.2 Abstract Translation

In the abstract translation task (Table 3b) the overall performance of the models in terms of BLEU scores has been considerably lower. This is understandable, as we did not have access to any in-domain parallel data for training. The baseline model using only the OOD parallel data has achieved an 8.67 BLEU score.

Nevertheless, we have been able to improve this result by applying our backtranslation and pretrained LMs. Just adding the backtranslated sen-

[2] https://github.com/bert-nmt/bert-nmt
[3] https://github.com/pytorch/fairseq

Training Data	Models									
	baseline		bert-base-uncased		bert-pubmed		bert-mimic-pubmed		bert-discharge-summaries	
	Accuracy	BLEU	Accuracy	BLEU	Accuracy	BLEU	Accuracy	BLEU	Accuracy	BLEU
ICD-10 train	73.15	88.70	**74.93**	**89.49**	74.60	89.39	74.67	89.31	74.78	89.42

(a) Terminology translation. Average results of 3 independent runs.

Training Data	Models				
	baseline	bert-base-uncased	bert-pubmed	bert-mimic-pubmed	bert-discharge-summaries
OOD Parallel	8.67	9.36	9.92	9.84	9.55
+ backtranslated general	11.57	14.34	13.71	**14.83**	13.97
+ backtranslated biomedical and ICD-10 train/dev	13.91	14.25	12.39	13.42	11.87

(b) Abstract translation. Average results of 3 independent runs.

Table 2: Results over the validation sets.

tences from Wikipedia (*backtranslated general* in Table 2) to the training set has improved the baseline by $+1.90$ BLEU pp, and adding the biomedical domain backtranslated sentences (*backtranslated biomedical*) and the ICD-10 parallel data has achieved a further $+2.41$ BLEU pp.

Additionally, we have observed comparable improvements with the BERT-fused NMT models, which have again consistently outperformed the baseline. In this case, the best performing pretrained LM has been the *bert-mimic-pubmed* model (14.83 BLEU) trained using only the *backtranslated general* data. This model has achieved an incremental improvement of $+6.16$ BLEU pp with respect to the baseline trained only over the OOD parallel data. It is interesting to see that adding the *backtranslated biomedical* data to the BERT-fused NMT model has not resulted in any improvement, probably because the data are more "noisy" (not grammatically well-structured sentences) and have fewer samples than the *general* backtranslations ($\sim 60,000$ vs 1.5 M).

5.3 Results over the Blind Test Sets

Table 3 shows the results achieved by our best performing models over the test sets. The translations made by our models have been submitted "blindly" and the results have been computed by the organizers. Our proposed runs for the terminology translation task have performed similarly to the validation set, achieving over 73% accuracy. On the contrary, the systems submitted to the abstract translation task have underperformed compared to the results in the validation set. We speculate this is likely due to the domain differences between our validation data and the test data. Even though both datasets are composed of translations of biomedical abstracts, they are probably coming from different databases and may have significantly different writing styles.

Model	Accuracy
bert-mimic-pubmed (run 1)	73.00
bert-discharge-summaries (run 2)	73.00
bert-base-uncased (run 3)	73.00

(a) Terminology translation.

Model	BLEU
bert-mimic-pubmed (run 1)	5.30
bert-pubmed (run 2)	**5.49**
baseline (run 3)	5.28

(b) Abstract translation.

Table 3: Official results over the blind test sets.

6 Conclusion

This work has described the translation systems submitted by the UTS_NLP team to the WMT20 Biomedical Translation shared task. The proposed systems are BERT-fused NMT models trained on a combination of in-domain parallel data, out-of-domain parallel data and backtranslations of monolingual data. The experiments have shown that combining pretrained BERT LMs and backtranslations during training has contributed to achieve considerable accuracy improvements with respect to a standard transformer-based NMT model trained only on the parallel data. Nevertheless, the official test results have shown that the performance of the systems can significantly drop if the translation domain is different. Therefore, there is still significant work to do to improve the domain adaption of these models.

References

Rodrigo Agerri, Iñaki San Vicente, Jon Ander Campos, Ander Barrena, Xabier Saralegi, Aitor Soroa, and Eneko Agirre. 2020. Give your text representation models some love: the case for basque. In *Proceedings of the Language Resources and Evaluation Conference*.

Emily Alsentzer, John R Murphy, Willie Boag, Wei-Hung Weng, Di Jin, Tristan Naumann, and Matthew McDermott. 2019. Publicly available clinical bert embeddings. In *Proceedings of the Clinical Natural Language Processing Workshop*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate.

Franck Burlot and François Yvon. 2018. Using monolingual data in neural machine translation: a systematic study. In *Proceedings of the Third Conference on Machine Translation*.

Stéphane Clinchant, Kweon Woo Jung, and Vassilina Nikoulina. 2019. On the use of bert for neural machine translation. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the North American Chapter of the Association for Computational Linguistics*.

Sergey Edunov, Alexei Baevski, and Michael Auli. 2019. Pre-trained language model representations for language generation. In *Proceedings of the North American Chapter of the Association for Computational Linguistics*.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-Wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. 2016. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3(1):1–9.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the International Conference for Learning Representations*.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. In *Proceedings of the Conference on Neural Information Processing Systems*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the Association for Computational Linguistics*, pages 311–318.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the North American Chapter of the Association for Computational Linguistics*.

A Poncelas, D Shterionov, A Way, GM de Buy Wenniger, and P Passban. 2018. Investigating backtranslation in neural machine translation. *arXiv preprint arXiv:1804.06189*.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *URL https://s3-us-west-2.amazonaws.com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. In *Proceedings of the Association for Computational Linguistics*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the Association for Computational Linguistics*.

Xabier Soto, Dimitar Shterionov, Alberto Poncelas, and Andy Way. 2020. Selecting backtranslated data from multiple sources for improved neural machine translation. In *Proceedings of the Association for Computational Linguistics*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Jiacheng Yang, Mingxuan Wang, Hao Zhou, Chengqi Zhao, Yong Yu, Weinan Zhang, and Lei Li. 2020. Towards making the most of bert in neural machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Jinhua Zhu, Yingce Xia, Lijun Wu, Di He, Tao Qin, Wengang Zhou, Houqiang Li, and Tie-Yan Liu. 2020. Incorporating bert into neural machine translation. In *Proceedings of the International Conference on Learning Representations*.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27.

Lite Training Strategies for Portuguese-English and English-Portuguese Translation

Alexandre Lopes[1] **Rodrigo Nogueira**[2,3,4] **Roberto Lotufo**[2,3] **Helio Pedrini**[1]

[1]Institute of Computing, University of Campinas, Brazil
[2]School of Electrical and Computer Engineering, University of Campinas, Brazil
[3]NeuralMind Inteligência Artificial, Brazil
[4]David R. Cheriton School of Computer Science, University of Waterloo, Canada

Abstract

Despite the widespread adoption of deep learning for machine translation, it is still expensive to develop high-quality translation models. In this work, we investigate the use of pre-trained models, such as T5 (Raffel et al., 2019) for Portuguese-English and English-Portuguese translation tasks using low-cost hardware. We explore the use of Portuguese and English pre-trained language models and propose an adaptation of the English tokenizer to represent Portuguese characters, such as diaeresis, acute and grave accents. We compare our models to the Google Translate API and MarianMT on a subset of the ParaCrawl dataset, as well as to the winning submission to the WMT19 Biomedical Translation Shared Task. We also describe our submission to the WMT20 Biomedical Translation Shared Task. Our results show that our models have a competitive performance to state-of-the-art models while being trained on modest hardware (a single 8GB gaming GPU for nine days). Our data, models and code are available at https://github.com/unicamp-dl/Lite-T5-Translation.

1 Introduction

With the advent of deep neural networks, results in machine translation have recently improved over classical statistical strategies (Wu et al., 2016; Artetxe et al., 2018). For instance, in the Third and Fourth Conference on Machine Translation (WMT18 (Edunov et al., 2018) and WMT19 (Ng et al., 2019)), the top-performing systems for English-German and German-English competitions were based on transformers (Vaswani et al., 2017).

Transformer models are state-of-the-art architectures for Machine Translation (MT) tasks and are capable of translating the same word to different words based on the context. For instance, the word 'bank' in Portuguese can be translated to 'bench' or 'bank' depending on the context.

This work explores translation strategies using language models pre-trained on Portuguese and English corpora. More specifically, we investigate the use of **T**ext-to-**T**ext **T**ransfer **T**ransformer (T5) pre-trained model for these tasks. T5 is a text-to-text Transformer trained with a similar masked language modeling objective as BERT. In this model, all target tasks are cast as sequence-to-sequence tasks. An illustration of T5 for the English-Portuguese translation task is shown in Figure 1. The main contributions of this work are:

- We show that it is possible to train translation models that are competitive with the state of the art using few computational resources. We trained our models on a gaming desktop with an Nvidia RTX2070 GPU, i5 CPU, and 32GB RAM. In comparison, the winning submission of the WMT19 Biomedical Translation Shared Task used four NVIDIA V100 GPUs, each being approximately ten times more expensive than an RTX2070.
- We created and made public ParaCrawl 99k, a dataset of 99k sentence pairs extracted from ParaCrawl's v6.0 English-Portuguese parallel corpus (Bañón et al., 2020). This large test corpus allows researchers to evaluate their models on a general-domain translation task.
- We evaluated Google Translate on ParaCrawl 99k, allowing other researchers to compare their results to a high-quality commercial system.
- We developed an adaptation for the English pre-trained tokenizer and achieved better results on English-Portuguese translation tasks than using the tokenizer without any changes. This allows us to efficiently adapt language models to a vocabulary that was not seen during pre-training.

Proceedings of the 5th Conference on Machine Translation (WMT), pages 833–840
Online, November 19–20, 2020.

2 Related Work

The winning system of WMT'19 Biomedical competition for en-pt and pt-en translation tasks (Soares and Krallinger, 2019a) is a Neural Machine Translation (NMT) system. They used OpenNMT-py to train a transformer model on seven parallel corpora. However, differently from our models, their model was trained from scratch.

Recent works (Peters et al., 2018; Devlin et al., 2018) have shown the advantages of using pre-trained models for tasks such as question-answering and text classification. The intuition is to allow the network to use information from pre-training language representations to increase the performance on specific tasks.

Edunov et al. (2019) evaluated the use of a pre-trained encoder-decoder model for translation. Both encoder and decoder weights were tied, but they were pre-trained on different languages. This is an expensive strategy for techniques that use a trainable tokenizer, such as SentencePiece (Kudo and Richardson, 2018), because it is necessary to re-train the entire model if the vocabulary changes, as new token embeddings need to be learned.

Many commercial systems, such as Google Translate (GT) and Amazon Translate (AT), have an excellent performance on MT, but they are expensive if one needs to translate vast amounts of text. For example, we estimate that it would cost 50,000 USD to translate the 20 million sentences of ParaCrawl using GT. Unfortunately, no commercial system that we are aware of provides metric scores on public datasets that would allow us to compare their systems to ours.

3 Methods

We proposed two main strategies for translating: using a T5 model pre-trained on a Portuguese corpus and adapting the original T5 tokenizer to work with Portuguese texts.

3.1 Pre-trained Language Model

We evaluated four different scenarios: English-Portuguese translation with T5 pre-trained on a Portuguese corpus; English-Portuguese translation with T5 pre-trained on an English corpus; Portuguese-English translation with T5 pre-trained on an English corpus; Portuguese-English translation with T5 pre-trained on a Portuguese corpus.

These variations allow us to evaluate how the language used during pre-training affects the translation's performance.

3.2 Adaptation of the English tokenizer to Portuguese

Here we investigate if we can adapt to the English-Portuguese translation task a model already pre-trained on languages other than Portuguese.

We observe that using a non-Portuguese tokenizer can cause translation problems, since some Portuguese characters cannot be represented, such as letters with the tilde accent (e.g. 'ã'). To fix this issue, we propose an adaptation of the original T5 tokenizer using a pre-processing and post-processing strategy. The tokenizer's adaptation allows it to represent all possible characters in the Portuguese language.

We can divide this adaptation into two stages: *Token Completion* and *Word Regrouping*. The first stage allows the use of Portuguese special characters, such as accented vowels, whereas the second stage merges these extra tokens back to form correct words.

3.2.1 Token Completion Stage

In this step, we start adding to the tokenizer Portuguese accented vowels that were not present in it. We ended up adding fourteen of those characters, as well as the word 'não', which is the most common word in the ParaCrawl pt-en dataset.

A list of all added tokens is available in Table 1. The addition of these tokens allowed the model to learn and generate them in en-pt translation.

This is also an inexpensive method for increasing the number of words that can be represented since only the embeddings of the new tokens have to be learned from scratch. The existing token embeddings, which represent the majority of the non-Portuguese tokens, were already learned during the pre-training phase and can be reused in the fine-tuning phase.

We show in Table 2 some encoding and decoding examples after adding tokens to the tokenizer.

ì ò Á Í Ó Ú í ú Â Ê Ã Õ ã õ não

Table 1: List of tokens added to the T5 tokenizer by our adaption method.

3.3 Word Regrouping Stage

When adding tokens directly to the tokenizer, the HuggingFace's (Wolf et al., 2019) SentencePiece

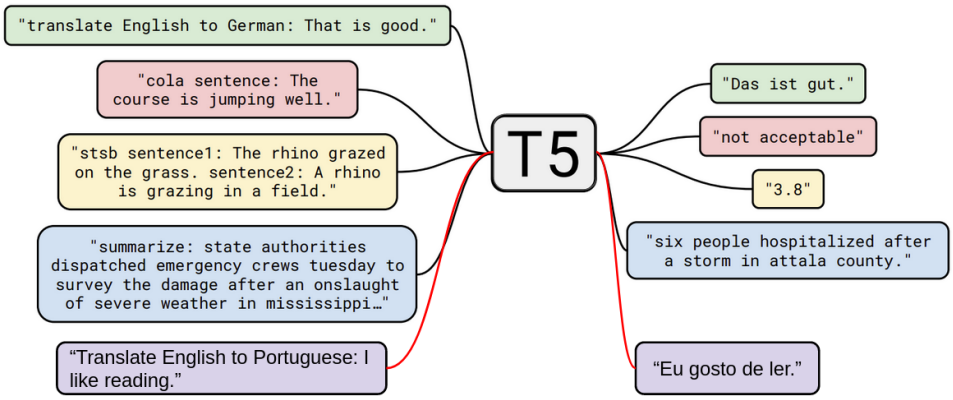


Figure 1: The text-to-text framework used by T5. The purple boxes and red connections represent the task used in this work. Figure adapted from (Raffel et al., 2019).

Tokenizer without additional Port. tokens
original → after encoding/decoding
eu gosto de arroz → eu gosto de arroz
eu não como → eu n ? o como
indignação completa → indignaç ? o completa
Tokenizer with additional Port. tokens
original → after encoding/decoding
eu gosto de arroz → eu gosto de arroz
eu não como → eu não como
indignação completa → indignaç ã o completa

Table 2: Comparing tokenizer results before and after adding the Portuguese tokens.

implementation used in our work interprets the result as a new complete token, i.e., not part of a word. For example, the word ’pão’ is broken into three different tokens ’p’ ’ã’ ’o’.

In this step, we regroup the added tokens of vowels with accents separated erroneously by the tokenizer. We find in the translated text all tokens added in the Token Completion step, and merge them with their neighboring words.

In Figure 2, we illustrate our algorithm.

4 Datasets

We trained our models using six different datasets, and we evaluate our system on two datasets: the WMT19 Biomedical Translation Task dataset and a subset of 99,000 sentence pairs of the ParaCrawl dataset. We also present the results of our submission to the WMT20 Biomedical Translation Task competition.

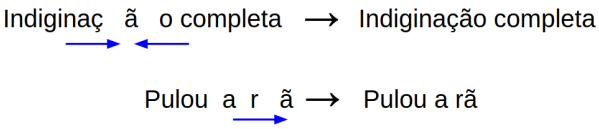


Figure 2: An example of separated tokens merged back into a single word. Our algorithm searches for an isolated special token (in this case, ’ã’) and merges it with its neighbors. It can be merged at the beginning, middle, or end of a sentence.

4.1 Training Datasets

We have two different strategies for training our models depending on the test datasets. For the evaluation on the ParaCrawl dataset, we only trained the models on ParaCrawl data. ParaCrawl is a public parallel corpus of many European languages available online. Its v6.0 version contains approximately 20M English-Portuguese sentence pairs. Due to our small computational budget, we randomly selected approximately 5M pairs for training.

For WMT19 and WMT20 Biomedical Translation Tasks, we train our models on the ParaCrawl dataset as well as on the following datasets, which are mostly of the same domain as WMT’s Biomedical data:

- EMEA Corpus (Tiedemann, 2012): A parallel corpus of European Medicines Agency documents.
- CAPES Parallel Dataset (Soares et al., 2018b): A parallel corpus of theses and dissertations abstracts collected from the CAPES website.
- Scielo Dataset (Soares et al., 2018a): A parallel corpus of scientific articles collected from

SciELO.

- JRC-Acquis (Steinberger et al., 2006): A parallel corpus of European Union (EU) documents in all official EU languages.

- Biomedical Domain Parallel Corpora (Névéol et al., 2018): A repository of the challenge that contains links to different parallel corpora. We used the Medline, Scielo, and ReBEC training datasets.

Besides being of the same domain as WMT's Biomedical task, an advantage of these datasets over ParaCrawl is that they are in Brazilian Portuguese, such as most of WMT's Biomedical data. The number of sentence pairs used for training from each dataset is shown in Table 3.

Corpus	Sent. Pairs
EMEA	1,082,144
CAPES	1,157,610
Scielo	2,828,916
JRC-Acquis	1,236,846
Biomedical Domain Corpora	331,937
Total	**6,637,453**

Table 3: Number of sentence pairs of each domain-specific dataset used to train our models for the WMT19 and WMT20 Biomedical tasks.

4.2 Testing Datasets

We created a general-domain test set from the ParaCrawl dataset. We begin by randomly selecting 128,000 sentence pairs from its 20M pairs. ParaCrawl is originally deduplicated, but similar sentences still might exist in our split of the training and test sets. Thus, we apply as stricter deduplication process to increase the quality of our test set. To increase the speed in verifying similarity of sentence pairs, we used MinHash and Locality-Sensitive Hashing (LSH) (Rajaraman and Ullman, 2011) to compare sentences of training and test datasets. We set a Jaccard similarity threshold to 0.7, i.e., all sentences with similarity greater than 0.7 were discarded from the test set. LSH found 28,913 sentences in the test set with a similarity score above 0.7 of sentences in the training set. The final test set ended up having 99,087 sentence pairs, which we called ParaCrawl 99k test set. This dataset and its corresponding translations using GT are available in our GitHub repository.

We also evaluated our system on the WMT19 Biomedical Shared Task test set. This is a dataset composed of approximately 500 parallel sentences of Medline abstracts.

Finally, we submitted our results to the WMT20 Biomedical Shared Task competition. The WMT20 test set has 544 parallel sentences for the English-Portuguese translation task and 498 sentences for the Portuguese to English task.

5 Experiments

We conducted several experiments using different configurations of T5. We divided the experiments into two groups: model hyperparameter optimization and different pre-training studies. All experiments were performed on a desktop computer with an Nvidia 8GB RTX 2070 Super, 32 Gb RAM memory, and a 4-core Intel processor running on Ubuntu 18.04. We used PyTorch (Paszke et al., 2017), HuggingFace Transformer, and Pytorch-Lightning (Falcon, 2019) frameworks to train and evaluate our models.

5.1 Model Hyperparameter Optimization

We tuned the hyperparameters using the original T5 checkpoint available in the HuggingFace library. This model was pre-trained on a corpus whose majority of documents were in English with a small proportion of German, French, and Romanian documents. We first conducted a small training using 100k sentence pairs and evaluated on another 50k sentence pairs to determine some hyperparameters of the T5 model, such as batch size and the maximum length of tokens in the source and target sentences. We also evaluated the optimizer and found the best convergence with the AdamW Optimizer (Loshchilov and Hutter, 2017). All hyperparameters used are in Table 4. With this configuration, we evaluated the performance of adding Portuguese-only characters to the tokenizer in comparison to using the original T5 tokenizer. The results are available in Table 5. Our proposed tokenizer adaption resulted in an improvement of almost 5 BLEU points over the original tokenizer in the en-pt translation task. All BLEU scores reported in this paper were generated using SacreBLEU (Post, 2018) with "intl" tokenization.

After finding these hyperparameters, we analyzed the trade-off between model sizes in a subset of the ParaCrawl dataset of 1M sentence pairs and evaluated them in a 150k sentence subset. We did

Hyperparameters	Values
Batch Size	256
Source Sequence Length (SSL)	96
Target Sequence Length (TSL)	160
Learning Rate	$5 \cdot 10^{-3}$
eps	$1 \cdot 10^{-5}$

Table 4: Hyperparameters used for training the models.

Translation Type	BLEU
Original T5 tokenizer	31.15
+ Portuguese characters	35.95 (+4.8)

Table 5: Effects in performance of using our adaption of the original T5 tokenizer to the English-Portuguese translation task. Numbers are from ParaCrawl's 99k en-pt test set.

not use any sentence from the test set. The results of this analysis are reported in Table 6. We trained the T5-small and T5-base models with different epoch sizes. Training 3 epochs of T5-small takes almost the same time as training one epoch with a T5-base model.

The performance would possibly increase if we used large models such as T5-large, T5-3B, or T5-11B. However, we could fit only the T5-base model in our 8GB GPU. We used batch accumulation to achieve batches of size 256 as the T5-small can only handle batch size 4 in 8GB. Thus, one of the contributions of this work is to show that it is possible to train translation models that are close to the state of the art on a relatively inexpensive hardware.

We also conducted experiments changing SSL and TSL lengths. We found that it is not necessary to set large TSL and SSL values, e.g., 256 for both, but it is essential to choose a value that can represent the distribution of target and source sequences. In later sections, we conduct experiments with TSL and SSL 140 and 160, respectively, since 99.8 % of our training dataset is shorter than these values.

All experiments in the following sections using Tokenizer's Adaptation Steps (3.2) were performed using the best pre-processing and post-processing strategies presented in Table 6.

5.1.1 Pre-training Studies

We also evaluated the effects of pre-training the model in a corpus of the same language of the target language. The intuition here is that it would be easier for the model to learn the target language than having previous knowledge of the source language. Since the tokenizer mainly has tokens of one of the two languages, it is better to have a smaller quantity of tokens to learn. This is because, if the SentencePiece tokenizer does not have the word in its vocabulary, it will use subtokens to form the original word. For example, the sentence 'They like to drink coconut water' is represented by six tokens in English SentencePiece and thirteen tokens in Portuguese SentencePiece. We are not evaluating here the possibility to train the pre-training model from scratch with both languages together, as it is not possible with our modest hardware setup.

Translation Type	Sacre BLEU
Adding Top 25 words in Port. + T5-small + 3 epochs	43.03
Adding tokens of Table 1 in Port. + T5-small + 3 epochs	43.48
Adding tokens of Table 1 in Port. + T5-base + 1 epoch	**44.52**

Table 6: Effects in performance of different strategies for adapting the original T5 tokenizer to Portuguese. Numbers are from our dev set of ParaCrawl.

For the Portuguese pre-trained model, we used PTT5-base model (Carmo et al., 2020) with Portuguese tokenizer. PTT5 was pre-trained on BrWAC, a large corpus of Brazilian Portuguese webpages. PTT5 started training using T5's official published weights as initial weights, so it also uses English learning in its model. For the English pre-trained model, we used the Huggingface implementation of T5 with its default tokenizer, which is based on SentencePiece.

In Table 7, we compare both models with Google Translate in the ParaCrawl 99k test set. Both models perform similarly in the Portuguese-English translation task, but the Portuguese pre-trained model gives a better result than the English pre-trained model in the English-Portuguese translation task. We are on par with Google Translate on en-pt, but a few BLEU points below on pt-en.

6 WMT19 and WMT20 Results

We now evaluate our models on the WMT19 Biomedical Translation Task and the results of our best models and official submission to the WMT20 Biomedical Translation Task.

	pt-en	en-pt
Google Translate API	51.20	45.17
Ours - English pre-training	46.49	44.56
Ours - Portuguese pre-training	46.35	45.44

Table 7: BLEU comparison between GT and our approach in Paracrawl 99k test set.

In Table 8, we show WMT19 results of our models as well as the winning submission of WMT19 Biomedical tasks (Soares and Krallinger, 2019b) and the MarianMT (Junczys-Dowmunt et al., 2018) implementation available on the HuggingFace's Transformer Library.[1] MarianMT's translation uses a multilingual Romance model, which is only possible to set the target language. This explains MarianMT's low performance on Portuguese to English translation since the model has to infer that the input sentence is in Portuguese. Models pre-trained on Portuguese obtained the best performance in both translation tasks. Notably, we achieved an improvement of +6.31 BLEU points in the English to Portuguese translation task by using the Portuguese pre-trained model and +9.75 with an increase of target and source sequence lengths. We also obtained an improvement of +0.62 in the Portuguese to English translation task using the Portuguese pre-trained model and +2.27 when increasing target and source sequence lengths.

We believe that the improvement of Portuguese pre-training models is associated with PTT5's training strategy that uses English pre-trained weights as initial weights. The intuition is that PTT5 carries information from the English model too.

The results for WMT20's challenge are in Table 9. Our submission is 2.17 BLEU points below the winning team in Portuguese-English, but it is 4.48 BLEU points higher than the baseline. For the English-Portuguese task, our results are below the baseline. That can be attributed to not using the Portuguese pre-trained model, which was not available at the time of our submission. As noted above, we achieved a large improvement on WMT19 when we switched from the English to the Portuguese pre-trained model. Therefore, we assume that a Portuguese pre-trained model would obtain superior results to the baseline on WMT20.

We also evaluated our Portuguese pre-training models with the best participants submissions of the WMT20 Biomedical Shared Task. The results of this comparison are in Table 10 and are slightly different from Table 9 since we are using SacreBLEU as the evaluation script. Our best translation models are 1.3-1.5 BLEU points bellow the winning submission (Sheffield) in both translation directions (pt-en and en-pt).

[1] https://huggingface.co/transformers/model_doc/marian.html

	pt-en	en-pt
MarianMT	27.91	47.44
BSC (Soares and Krallinger, 2019b)	39.90*	48.18*
Ours - English pre-training	45.89	39.31
Ours - Portuguese pre-training	46.51	45.62
+ TSL=256 and SSL=256	–	**49.06**
+ TSL=140 and SSL=160	**48.16**	–

Table 8: BLEU scores on the test set of WMT19 Biomedical Shared Task. Portuguese pre-training was tested in three different scenarios: one with default hyperparameters available in Table 4 and two with different Target Sequence Length (TSL) and Source Sequence Length (SSL). *This is the official submission score.

Team Names	pt-en	en-pt
Sheffield	48.16	44.57
Unicamp_DL	45.99*	38.08*
baseline	41.51	39.77

Table 9: BLEU scores on WMT20's automatic evaluation. *Since the Portuguese T5 model was not available at the time of our submission, we used the original (English) T5. Hence, results for en-pt and pt-en could be improved by switching to the Portuguese pre-trained model.

Team Names	pt-en	en-pt
Unicamp_DL	48.67	40.18
Sheffield	**52.25**	**46.69**
Ours - Portuguese pre-training		
+ TSL=256 and SSL=256	–	45.33
+ TSL=140 and SSL=160	50.75	–

Table 10: BLEU scores on the test set of WMT20 Biomedical Shared Task.

7 Conclusions and Future Work

We show that it is possible to develop English-Portuguese translation models close to the state of the art using modest hardware. Despite not reaching the same level of performance of Google Translate on pt-en, the fact that our system was devel-

oped mostly by the first author on its personal computer shows that implementing high-quality machine translation systems has become possible for anyone, including small companies and research labs. We also cannot guarantee that Google did not use our testing data for training.

We presented our submission strategies for the WMT20 Biomedical Translation Shared Task using a T5 model. We show that a simple adaption of the original T5 tokenizer to the Portuguese language largely improves the translation quality and does not require any further pre-training, which is expensive. However, we achieve the best results with models pre-trained on Portuguese.

As directions for future work, we plan to experiment with larger models and models pre-trained on both Portuguese and English languages simultaneously, as recent work showed that this a successful strategy (Wu et al., 2016; Arivazhagan et al., 2019). We believe that we could improve the translation results with larger and more complex models (Lepikhin et al., 2020).

8 Acknowledgements

We thank CNPq research funding, process number 310828/2018-0.

References

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George Foster, and Colin Cherry. 2019. Massively multilingual neural machine translation in the wild: Findings and challenges. *arXiv preprint arXiv:1907.05019*.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018. Unsupervised statistical machine translation. In *Conference on Empirical Methods in Natural Language Processing*, pages 3632–3642, Brussels, Belgium. Association for Computational Linguistics.

Marta Bañón, Pinzhen Chen, Barry Haddow, Kenneth Heafield, Hieu Hoang, Miquel Esplà-Gomis, Mikel L. Forcada, Amir Kamran, Faheem Kirefu, Philipp Koehn, Sergio Ortiz Rojas, Leopoldo Pla Sempere, Gema Ramírez-Sánchez, Elsa Sarrías, Marek Strelec, Brian Thompson, William Waites, Dion Wiggins, and Jaume Zaragoza. 2020. ParaCrawl: Web-scale acquisition of parallel corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4555–4567, Online. Association for Computational Linguistics.

Diedre Carmo, Marcos Piau, Israel Campiotti, Rodrigo Nogueira, and Roberto Lotufo. 2020. Ptt5: Pre-training and validating the t5 transformer in brazilian portuguese data. https://github.com/unicamp-dl/PTT5.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, pages 1–16.

Sergey Edunov, Alexei Baevski, and Michael Auli. 2019. Pre-trained language model representations for language generation. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4052–4059, Minneapolis, Minnesota. Association for Computational Linguistics.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Conference on Empirical Methods in Natural Language Processing*, pages 489–500, Brussels, Belgium. Association for Computational Linguistics.

WA Falcon. 2019. Pytorch lightning. *GitHub. Note: https://github.com/PyTorchLightning/pytorch-lightning Cited by*, 3.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, and Nikolay Bogoychev. 2018. Marian: Fast neural machine translation in C++. *arXiv preprint arXiv:1804.00344*, pages 1–6.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. 2020. Gshard: Scaling giant models with conditional computation and automatic sharding. *arXiv preprint arXiv:2006.16668*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Aurélie Névéol, Antonio Jimeno Yepes, Mariana Neves, and Karin Verspoor. 2018. Parallel corpora for the biomedical domain. In *Eleventh International Conference on Language Resources and Evaluation*, Miyazaki, Japan. European Language Resources Association (ELRA).

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook FAIR's WMT19 news translation task submission.

In *Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 314–319, Florence, Italy. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, pages 1–67.

Anand Rajaraman and Jeffrey David Ullman. 2011. *Mining of Massive Datasets*. Cambridge University Press.

Felipe Soares and Martin Krallinger. 2019a. BSC participation in the WMT translation of biomedical abstracts. In *Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 175–178, Florence, Italy. Association for Computational Linguistics.

Felipe Soares and Martin Krallinger. 2019b. BSC Participation in the WMT Translation of Biomedical Abstracts. In *Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 175–178.

Felipe Soares, Viviane Moreira, and Karin Becker. 2018a. A large parallel corpus of full-text scientific articles. In *Eleventh International Conference on Language Resources and Evaluation*.

Felipe Soares, Gabrielli Harumi Yamashita, and Michel Jose Anzanello. 2018b. A parallel corpus of theses and dissertations abstracts. In *International Conference on Computational Processing of the Portuguese Language*, pages 345–352. Springer.

Ralf Steinberger, Bruno Pouliquen, Anna Widiger, Camelia Ignat, Tomaž Erjavec, Dan Tufiş, and Dániel Varga. 2006. The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages. In *Fifth International Conference on Language Resources and Evaluation*, Genoa, Italy. European Language Resources Association (ELRA).

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *Eight International Conference on Language Resources and Evaluation*, Istanbul, Turkey. European Language Resources Association (ELRA).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

The ADAPT's Submissions to the WMT20 Biomedical Translation Task

Prashanth Nayak, Rejwanul Haque and Andy Way
The ADAPT Centre, School of Computing
Dublin City University, Dublin, Ireland
firstname.lastname@adaptcentre.ie

Abstract

This paper describes the ADAPT Centre's submissions to the WMT20 Biomedical Translation Shared Task for English-to-Basque. We present the machine translation (MT) systems that were built to translate scientific abstracts and terms from biomedical terminologies, and using the state-of-the-art neural MT (NMT) model: Transformer. In order to improve our baseline NMT system, we employ a number of methods, e.g. "pseudo" parallel data selection, monolingual data selection for synthetic corpus creation, mining monolingual sentences for adapting our NMT systems to this task, hyperparameters search for Transformer in low-resource scenarios. Our experiments show that systematic addition of the aforementioned techniques to the baseline yields an excellent performance in the English-to-Basque translation task.

1 Introduction

The ADAPT Centre participated in the Biomedical Translation Shared Task of the Fifth Conference of Machine Translation (WMT20). This task is about evaluating systems on the translation of documents from the biomedical domain. The test data consists of biomedical abstracts and terminologies. The task addresses a number of language pairs, and we participated in the English-to-Basque translation task. To make the readers familiar with the biomedical translation task and to understand the challenges of this task, we show a couple of examples from the blind test set and two terminological expressions from terminology test set in Table 1.

For building our MT systems we used the Transformer model (Vaswani et al., 2017). Our strategies to build the competitive MT systems for the task roughly include (i) pseudo in-domain parallel and monolingual data selection, (ii) augmenting training data (Sennrich et al., 2016a; Zhang and Zong, 2016; Burlot and Yvon, 2018; Poncelas et al., 2018; Caswell et al., 2019; Chen et al., 2019), (iii) mining monolingual sentences to adapt our NMT systems to the task, and (iv) finding the optimal hyperparameter configuration for Transformer in this low-resource settings.

(1) No cardiovascular risk factor differences were found in terms of age.
(2) Congenital tumors show a different pattern than tumors in other pediatric ages.
(3) Open bite of thyroid gland, sequela
(4) poisoning by oxytocic drugs, undetermined, subsequent encounter

Table 1: Sentences ((1) and (2)) from the blind test set and sample terminological expressions ((3) and (4)).

The remainder of the paper is organized as follows. In Section 2, we present our methods, and Section 3 details of the data sets used. Section 4 presents the results and discussions, while Section 5 concludes our work with avenues for future work.

2 Our Approaches

2.1 Selecting pseudo In-domain Parallel Sentences

The shared task organisers released parallel training data with a limited number of in-domain examples (only 24,247). The organisers also provided the participants with moderate-sized three out-of-domain corpora (totalling to approximately 770K bitexts). In an attempt to improve the quality of our baseline MT systems, we extracted those sentence-pairs from the out-of-domain corpora that are similar to the styles and domain of the texts we aim to translate, and were used in system building.

2.1.1 Selection using Bilingual Cross-Entropy Difference

We followed the state-of-the-art sentence selection approach of Axelrod et al. (2011) that extracts pseudo in-domain sentences from out-of-domain

Proceedings of the 5th Conference on Machine Translation (WMT), pages 841–848
Online, November 19–20, 2020.

corpora using bilingual cross-entropy difference over each side of the corpus (source and target). The bilingual cross-entropy difference is computed by querying in- and out-of-domain (source and target) language models.

2.1.2 Selection using Terminology

Terms are usually indicators of the nature of a domain and plays a critical role in domain-specific MT (Haque et al., 2020). Sentences that contain domain terms are likely to be a domain text. However, a ambiguous term could have more than one potential meaning. As an example of lexical ambiguity, 'cold' has several possible meanings in the Unified Medical Language System Metathesaurus (Humphreys et al., 1998) including 'common cold', 'cold sensation' and 'cold temperature' (Stevenson and Guo, 2010). We can see that 'cold' could have very different meanings depending on the context in which it appears. Moreover, a polysemous term (e.g. 'cold') could have many translation equivalents in a target language.

In our second sentence selection approach, we mine those sentences from large out-of-domain or general domain corpus that contain domain terms. As pointed out above, an extracted sentence that contain a domain term may not represent the desired domain; however, the training examples that include such extracted sentences may play crucial role in minimising lexical selection errors as far as terminology translation is concerned (Haque et al., 2020).

To this end, we exploit the approach of Rayson and Garside (2000) and Haque et al. (2014, 2018) in order to automatically identify terms in the in-domain texts. The idea is to identify those words which are most indicative (or characteristic) of the in-domain corpus compared to a reference corpus. Haque et al. (2014, 2018) used a large corpus which is generic in nature as a reference corpus. We adopted their approach and used a large generic corpus in order to identify terms in the in-domain source (English) and target (Basque) corpora. Given the lists of source and target terms, we mine sentences independently from the source- and target-sides of the out-of-domain bilingual corpus. We select those sentence-pairs from the out-of-domain bilingual corpus whose source or target sides contain at least one domain term.

2.2 Training Data Augmentation

The data augmentation methods in NMT (Sennrich et al., 2016a; Zhang and Zong, 2016; Burlot and Yvon, 2018; Bogoychev and Sennrich, 2019; Caswell et al., 2019; Chen et al., 2019), which usually employ the unlabeled monolingual data in addition to limited bitexts, can positively impact translation quality and are very popular among the MT developers and researchers (Barrault et al., 2019). In other words, use of synthetic data to improve a NMT system is nowadays a common practice, especially in the under-resource scenarios.

The synthetic training data whose source-side sentences are original is more effective for domain adaptation. The learning method that uses such training data is called self-training (Ueffing et al., 2007). The synthetic training data whose target-side is original is more effective for domain text translation and generation of fluent translations (Sennrich et al., 2016a). Many studies (e.g. Chen et al. (2019); Bogoychev and Sennrich (2019)) have shown that self-training and back-translation can be complementary to each other.

In this task, in order to improve our baseline Transformer models, we augmented our training data with both the target- and source-original synthetic data. As in Caswell et al. (2019), in order to let the NMT model know that the given source is synthetic, we tag the source sentences of the synthetic data with the extra tokens.

Iterative generation and training on synthetic data can yield increasingly better NMT systems, especially in low-resource scenarios (Hoang et al., 2018; Chen et al., 2019). Since our baseline source-to-target and target-to-source MT systems are already excellent in quality, those were used to translate the monolingual data.

As in Section 2.1, we extract those sentences from large monolingual data that are similar to the styles of texts we aim to translate. We used the extracted pseudo in-domain monolingual sentences to produce the source- and target-original synthetic bitexts. As for the NMT training, we believe that synthetic parallel data created from pseudo in-domain sentences could be the better alternatives than those selected randomly.

2.2.1 Selection using Language Model Perplexity

Sentences of a large monolingual corpus similar to the in-domain sentences when selected based on the perplexity according to an in-domain language model were found to be effective in MT (Gao et al., 2002; Yasuda et al., 2008; Foster et al., 2010; Axelrod et al., 2011; Toral, 2013). Accordingly, we select "pseudo" in-domain sentences from a large monolingual data based on their perplexity scores accoring to the in-domain language model, which

are then translated to form synthetic training data.

2.2.2 Selection using Terminology

We mine "pseudo" in-domain sentences from large monolingual corpora following the method described in Section 2.1.2. We select those sentences from the monolingual corpus that contain at least one domain term. For mining monolingual sentences we create an efficient Trie structure given the large monolingual data. The idea is to store indices of the sentences (i.e. we restrict this number to 50) for each *n*-gram (upto trigram) of the corpus. Given the domain terms of the in-domain text, we can instantly retrieve the sentences from corpus.

2.3 Mining Sentences for Fine-tuning

Chinea-Ríos et al. (2017) demonstrated that in case of specialised domains or low-resource scenarios where parallel corpora are scarce sentences of a large monolingual data that are more related to the test set sentences to be translated could be effective for fine-tuning the original general domain NMT model. They select those instances from large monolingual corpus whose vector-space representation is similar to the representation of the test set instances. The selected sentences are then automatically translated by an NMT system built on a general domain data. Finally, the NMT system is fine-tuned with the resultant synthetic data. In a similar line of research, it has also been shown that an NMT system built on general domain data can be fine-tuned using just a few sentences (Farajian et al., 2017, 2018; Wuebker et al., 2018; Huck et al., 2019).

2.3.1 Mining Source Language Monolingual Sentences

Since English–Basque is a low-resource language-pair and have a little amount of bitexts pertaining to the targeted domain (biomedical), we followed Chinea-Ríos et al. (2017) in order to mine those sentences from large monolingual data that could be beneficial for fine-tuning the original NMT models. In other words, we followed the method described in Section 2.1.2 in order to extract sentences form large monolingual corpus. As above, we identify terms in the test set (i.e. scientific abstracts of Medline) to be translated. As for the sub-task where the task is to translate the domain terms from English to Basque, we observed that many terminological entries are in fact a part of full sentences (e.g. 'person on outside of car injured in collision with pedestrian or animal in traffic accident, initial encounter') and contain general domain tokens. Therefore, we treat the terminological entries as normal sentences and translate them similarly to the Medline abstracts.

In addition to following the standard terminology extraction methods of Haque et al. (2014, 2018) who used a large corpus which is generic in nature as a reference corpus, in a second setup, we used either side of the authentic training bitexts on which the NMT systems were trained as the reference corpus. The intuition is to extract those terminological expressions from the test set that do not occur or rarely occur in the training data and are more indicative of the test corpus. We merged the two sets of terms extracted following the two setups above. Given the resultant list of terms, we mine sentences from monolingual corpus. The source sentences that have been mined are translated with the MT system in order to form synthetic bitexts to be used for adaptation.

2.3.2 Mining Bitexts

Farajian et al. (2017, 2018) exploit the similarity between the source sentences of the training examples and each test sentence and update their generic NMT model on-the-fly on a set of most similar training examples. Like them, we mine training examples form the bilingual training corpus. However, unlike them, our extraction process is driven by the domain terms appearing in the test set which is to be translated. In sum, we follow the bilingual sentence-pair extraction method described in Section 2.1.2 given the test set. For extraction we considered both in-domain and out-of-domain parallel corpora. The extracted bitexts are merged with the generated synthetic segment-pairs above (cf. Section 2.3.1). As in Chinea-Ríos et al. (2017), the best NMT system is finally fine-tuned on the combined train data.

2.4 Tuning Hyperparameters for Transformer

The NMT systems are Transformer models (Vaswani et al., 2017). To build our NMT systems, we used the MarianNMT (Junczys-Dowmunt et al., 2018) toolkit. The tokens of the training, evaluation and validation sets are segmented into sub-word units using Byte-Pair Encoding (BPE) (Sennrich et al., 2016b). We found that performance of the Transformer model more-or-less similar whether BPE is applied individually or jointly on the source and target languages. We kept the former setup, i.e. BPE is applied individually on the source and target languages. Recently, Sennrich and Zhang (2019) demonstrated that commonly used hyperparameter configuration do not lead to the best results in

low-resource settings. Accordingly, we carried out a series of experiments in order to find the best hyperparameter configuration for Transformer in our low-resource setting. In particular, we played with some of the hyperparameters, and found that the following configuration lead to the best results in our low-resource translation settings: (i) the BPE vocabulary size: 6,000, (ii) the sizes of the encoder and decoder layers: 4 and 6, respectively, and (iii) learning-rate: 0.0003. The models are trained with the Adam optimizer (Kingma and Ba, 2014), reshuffling the training corpora for each epoch. As for the remaining hyperparameters, we followed the recommended best setup from (Vaswani et al., 2017). The early stopping criteria is based on cross-entropy; however, the final NMT system is selected as per the highest BLEU score on the validation set. The beam size for search is set to 12. We make our final NMT model with ensembles of 8 models that are sampled from the training run.

3 Data Used

This section presents the data sets which were used for system building. We used the bilingual data provided by the WMT20 Biomedical Shared Task organisers only. As for English monolingual corpus, we used all in-domain texts released by the organisers including the English side of the bilingual corpora of the language-pairs. As for Basque monolingual data, organisers provided us with a tiny set of in-domain sentences. Since the participants are allowed to use external data, we used the CommonCrawl[1] corpus for Basque. Table 2 presents the corpus statistics. The out-of-domain

Bilingual			
in-domain	sentences	words (EN)	words (EU)
train	24,247	201,583	205,334
development	2,000	16,324	16,667
out-of-domain	770,273	12,637,438	11,289,811
Monolingual (sentences)			
	in-domain	CommonCrawl	
Basque	41,151	12,583,122	
English	9,015,051		

Table 2: The Corpus statistics.

parallel corpora for the English-to-Basque task are from three different sources (i.e OPUS (Tiedemann, 2012), IWSLT 2018 (Jan et al., 2018) and WMT16 IT Shared task (Bojar et al., 2016)). We merged segment-pairs of all three data sources, and after applying cleaning scripts to the data we are left with 770K parallel segments (cf. fifth row of Table 2).

[1] https://commoncrawl.org/

Since the size of English in-domain monolingual corpus is reasonably big, we did not use any English out-of-domain data for system building. In order to perform tokenisation for English and Basque texts, we used the standard tool of the Moses toolkit. The development data released by the task organisers contains 2,000 sentences (cf. fourth row of Table 2), out of which 1,000 sentences are used as the test set. The remaining sentences of the development set are used for validation.

4 Experiments and Results

This section presents the performance of our MT systems in terms of the automatic evaluation metric BLEU (Papineni et al., 2002). Additionally, we performed statistical significance tests using bootstrap resampling methods (Koehn, 2004).

4.1 The Baseline MT System

First, we build an English-to-Basque NMT system on the in-domain parallel corpus (cf. Table 2) only, and we refer the MT system as Base. Note that size of the original test set is 1,000 and its sentences were randomly sampled from development set released by the organisers (cf. Section 3). We evaluate Base on the original test set and report its BLEU score in Table 3. As far as the BLEU score on original test set is concerned, it is excessively high. When we looked at the translations, we saw that they are nearly perfect. We

	BLEU
Original test set (1,000)	91.12
test set (200)	47.14

Table 3: The BLEU scores of the basline NMT system (Base).

checked how similar the original test set sentences is to the in-domain training set sentences. For this, we apply fuzzy string matching with a threshold of 80%, and used SimString[2] algorithm (Okazaki and Tsujii, 2010) for search. We found that the number of the non-matching sentences of the test set is 200 (out of 1,000), and same of the development set is 194 (out of 1,000). This indicates that the test and development sets sentences are very similar to those of the training set. The scores on the original test and development sets could be misleading for the evaluation and validation of MT systems. Therefore, for fair evaluation we used the non-matching sentences as the test set (200).

[2] http://www.chokkan.org/software/simstring/

Note that the BLEU scores reported in this paper are on this test set. The BLEU scores of Base on the test set is reported in the last column of Table 3. Similarly, we used the non-matching sentences of the original development set as the development set (194).

4.2 The Improved MT Systems

We applied the pseudo in-domain bilingual sentence selection strategies described in Section 2.1 to the out-of-domain bilingual data (cf. Table 2). We first apply the bilingual cross-entropy difference (BCED) measure described in Section 2.1.1. The so-called pseudo in-domain parallel sentences that were extracted from the out-of-domain data were appended to the in-domain training data, and the BLEU scores of the NMT systems trained on the combined training data are shown in the top rows of Table 4. As can be seen from the table, when the size of pseudo in-domain data is 100K, the MT system (Base+BCED-100K) produces 50.68 BLEU on the test set (a 3.54 BLEU points corresponding to 7.5% relative gain over the Base).

	BLEU
Base+BCED-100K	50.68
Base+BCED-150K	49.02
Base+BCED-200K	47.38
Base+BiTerm	52.19
Base+BiTerm+BCED-100K	53.07

Table 4: The BLEU scores of the NMT systems trained on the in-domain added with the pseudo in-domain training data.

Next, we apply our second method (cf. Section 2.1.2), and the pseudo bilingual corpus extracted following this method contains 294,998 segment-pairs. As above, we append this data to the in-domain data. The BLEU score of MT system (Base+Term) built on the combined data is reported in Table 4. We see from the table that this strategy provides us a 5.05 BLEU points (corresponding to 10.7% relative) gain over the baseline.

When we merge these two pseudo in-domain parallel data with the real in-domain data and train the MT model on the combined data, we further achieved a moderate BLEU gain over the baseline (a 5.95 BLEU points corresponding to 12.6% relative gain). We used this MT system (Base+BiTerm+BCED-100K) for further experimentation, which, from now on, is referred to Base2.

	BLEU
Base2+BT1	52.72
Base2+BT2	53.65
Base2+BT3	53.70
Base2+FT1	52.02
Base2+FT2	51.45
Base2+BT3+FT1	52.76

Table 5: The BLEU scores of the NMT systems trained on augmented training data.

As pointed out above, we augment our bilingual training data with forward and back-translated synthetic data. The BLEU scores of the MT systems trained on the augmented training data are reported in Table 5.

First, we create a synthetic train data by back-translating the tiny monolingual in-domain training data, and the BLEU score of the MT system built on the training data that includes this synthetic data is shown in the second row of Table 5 (i.e. Base2+BT1). This data could not improve Base2.

We extract 275,125 sentences from Basque monolingual data following the method described in Section 2.2.2 (i.e. using the list of terminology extracted from in-domain corpus), and created synthetic bitexts as above. We further add these synthetic bitexts to the training data.[3] The BLEU score of the MT system trained on this data (Base2+BT2) is shown in Table 5. This MT system brings about a 0.58 BLEU points improvement over Base2, and this time, the improvement is not statistically significant.

We further select top 200K target sentences (Basque) based on perplexity scores following the method described in Section 2.2.1. Note that many extracted sentences overlap with those extracted using terminology. We obtained the similar BLEU score on the test set when the synthetic data that is created from this data is further appended to training data (i.e. Base2+BT3).

As mentioned above, we have large monolingual in-domain corpus for English (cf. Table 2). Therefore, we directly used the in-domain English sentences for self-learning. We carried out a number of experiments with adding the source-original synthetic sentences with the original training data, e.g. Base2+FT1 and Base2+FT2 refer to 200K and 1M synthetic segment-pairs. We started doing forward translation with the Medline text. The self-training strategy could not surpass the best-performing MT

[3] Note that this training data refers the one that corresponds to Base2+BT1.

system, i.e. Base2+BT3.

4.3 Fine-tuning the best NMT systems

This section presents the MT systems that were prepared by the adaptation technique described in Section 2.3. We select Base2+BT3 and Base2+BT3+FT1 for adaptation. Following the method described in Section 2.3.1 we mine the source monolingual sentences from the large English in-domain corpus given the list of terms extracted from the test set. Then, synthetic data is created by translating the source sentences by the source-to-target MT systems. We follow the method described in Section 2.3.2 and mine sentence-pairs from in- and out-of-domain bitexts given the list of terms extracted from the test set. The synthetic data and extracted sentence-pairs are merged to form training data for adaptation. Finally, the best MT systems were fine-tuned on this training data. The BLEU scores of the adapted MT systems on the test set are reported in Table 6. When we compare the original MT systems reported in Table 5 with the adapted MT systems, we see that (i) the adapted version of Base2+BT3 produces a 1.1 BLEU points (corresponding to 2.05% relative) improvement over Base2+BT3, and (ii) the same of Base2+BT3+FT1 produces a 1.51 BLEU points (corresponding to 2.87% relative) improvement over Base2+BT3+FT1. The improvements are statistically significant.

	BLEU
Base2+BT3	54.80
Base2+BT3+FT1	55.21

Table 6: The BLEU scores of the adapted MT systems.

As above, we create the adapted MT systems for the blind test set and terminology. Then, we translate the blind test set sentences and terminological entries with the adapted MT systems (Base2+BT3, Base2+BT3+FT1). For our third submission we chose a non-adapted MT system, Base+BiTerm+BCED-100K (cf. Table 4).

In Table 7, we show the BLEU scores of MT systems on the blind test sets. As for abstract translation, Base+BiTerm+BCED-100K is found to be the best system. This system earned us the third position in the task. For the evaluation of terminology translation, in addition to BLEU, the organisers used the accuracy metric which relies on strict matches between ground truth and predictions (cf. Table 7). Base2+BT3 and Base2+BT3+FT1 produce the best BLEU and accuracy scores, respectively, on the terminology test set. Our systems earned us the second position in the terminology translation task.

		BLEU
Base+BiTerm+BCED-100K		**8.67**
Base2+BT3 (adapted)		8.25
Base2+BT3+FT1 (adapted)		8.08
	Acc.	BLEU
Base+BiTerm+BCED-100K	0.73	70.83
Base2+BT3 (adapted)	0.75	**72.39**
Base2+BT3+FT1 (adapted)	**0.76**	71.79

Table 7: Performance of our submitted MT systems in the abstract (top 3 rows) and terminology (bottom 3 rows) translation tasks.

5 Conclusion

This paper presents the ADAPT system description for the WMT20 Biomedical Translation Shared Task. We participated in the English-to-Basque translation task. The task is to translate scientific abstracts and terms from biomedical terminologies. We aimed to build a competitive translation system for this task. For this, we applied various strategies, e.g. selecting monolingual and bilingual texts that are similar to the in-domain data, mining monolingual sentences, applying adaptation technique for adapting the neural MT models to the task, hyperparameters search. We found that our strategies to improve the baseline MT system were effective and yields excellent performance.

This paper demonstrated a novel adaptation approach for translating domain texts. This method is found to be effective in this translation task. In the future, we aim to test the on-the-fly adaptation method (Farajian et al., 2017, 2018) to translate domain texts.

Acknowledgments

The ADAPT Centre for Digital Content Technology is funded under the Science Foundation Ireland (SFI) Research Centres Programme (Grant No. 13/RC/2106) and is co-funded under the European Regional Development Fund. This project has partially received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 713567, and the publication has emanated from research supported in part by a research grant from SFI under Grant Number 13/RC/2077.

References

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. 2011. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 355–362, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Loïc Barrault, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. 2019. Findings of the 2019 conference on machine translation (wmt19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 1–61, Florence, Italy. Association for Computational Linguistics.

Nikolay Bogoychev and Rico Sennrich. 2019. Domain, translationese and noise in synthetic data for neural machine translation. *arXiv preprint arXiv:1911.03362*.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurelie Neveol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. 2016. Findings of the 2016 conference on machine translation. In *Proceedings of the First Conference on Machine Translation*, pages 131–198, Berlin, Germany.

Franck Burlot and François Yvon. 2018. Using monolingual data in neural machine translation: a systematic study. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 144–155, Belgium, Brussels. Association for Computational Linguistics.

Isaac Caswell, Ciprian Chelba, and David Grangier. 2019. Tagged back-translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 53–63, Florence, Italy. Association for Computational Linguistics.

Peng-Jen Chen, Jiajun Shen, Matthew Le, Vishrav Chaudhary, Ahmed El-Kishky, Guillaume Wenzek, Myle Ott, and Marc'Aurelio Ranzato. 2019. Facebook AI's WAT19 Myanmar-English translation task submission. In *Proceedings of the 6th Workshop on Asian Translation*, pages 112–122, Hong Kong, China.

Mara Chinea-Ríos, Álvaro Peris, and Francisco Casacuberta. 2017. Adapting neural machine translation with parallel synthetic data. In *Proceedings of the Second Conference on Machine Translation*, pages 138–147, Copenhagen, Denmark. Association for Computational Linguistics.

M. Amin Farajian, Nicola Bertoldi, Matteo Negri, Marco Turchi, and Marcello Federico. 2018. Evaluation of terminology translation in instance-based neural mt adaptation. In *Proceedings of the 21st Annual Conference of the European Association for Machine Translation*, pages 149–158, Alicante, Spain.

M Amin Farajian, Marco Turchi, Matteo Negri, and Marcello Federico. 2017. Multi-domain neural machine translation through unsupervised adaptation. In *Proceedings of the Second Conference on Machine Translation*, pages 127–137.

George Foster, Cyril Goutte, and Roland Kuhn. 2010. Discriminative instance weighting for domain adaptation in statistical machine translation. In *Proceedings of the 2010 conference on empirical methods in natural language processing*, pages 451–459.

Jianfeng Gao, Joshua Goodman, Mingjing Li, and Kai-Fu Lee. 2002. Toward a unified approach to statistical language modeling for chinese. *ACM Transactions on Asian Language Information Processing (TALIP)*, 1(1):3–33.

Rejwanul Haque, Mohammed Hasanuzzaman, and Andy Way. 2020. Analysing terminology translation errors in statistical and neural machine translation. *Machine Translation (in press)*, 34.

Rejwanul Haque, Sergio Penkale, and Andy Way. 2014. Bilingual termbank creation via log-likelihood comparison and phrase-based statistical machine translation. In *Proceedings of the 4th International Workshop on Computational Terminology (Computerm)*, pages 42–51, Dublin, Ireland.

Rejwanul Haque, Sergio Penkale, and Andy Way. 2018. TermFinder: log-likelihood comparison and phrase-based statistical machine translation models for bilingual terminology extraction. *Language Resources and Evaluation*, 52(2):365–400.

Vu Cong Duy Hoang, Philipp Koehn, Gholamreza Haffari, and Trevor Cohn. 2018. Iterative back-translation for neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 18–24, Melbourne, Australia. Association for Computational Linguistics.

Matthias Huck, Viktor Hangya, and Alexander Fraser. 2019. Better oov translation with bilingual terminology mining. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5809–5815.

Betsy L. Humphreys, Donald A. B. Lindberg, Harold M. Schoolman, and G. Octo Barnett. 1998. The Unified Medical Language System: An Informatics Research Collaboration. *Journal of the American Medical Informatics Association*, 5(1):1–11.

Niehues Jan, Roldano Cattoni, Stüker Sebastian, Mauro Cettolo, Marco Turchi, and Marcello Federico. 2018. The iwslt 2018 evaluation campaign. In *International Workshop on Spoken Language Translation*, pages 2–6.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 388–395, Barcelona, Spain.

Naoaki Okazaki and Jun'ichi Tsujii. 2010. Simple and efficient algorithm for approximate dictionary matching. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 851–859, Beijing, China.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *ACL-2002: 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, PA. ACL.

Alberto Poncelas, Dimitar Shterionov, Andy Way, Gideon Maillette de Buy Wenniger, and Peyman Passban. 2018. Investigating backtranslation in neural machine translation. In *Proceedings of The 21st Annual Conference of the European Association for Machine Translation (EAMT 2018)*, pages 249–258, Alicante, Spain.

Paul Rayson and Roger Garside. 2000. Comparing corpora using frequency profiling. In *The workshop on comparing corpora*, pages 1–6.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany.

Rico Sennrich and Biao Zhang. 2019. Revisiting low-resource neural machine translation: A case study. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 211–221, Florence, Italy. Association for Computational Linguistics.

Mark Stevenson and Yikun Guo. 2010. Disambiguation of ambiguous biomedical terms using examples generated from the umls metathesaurus. *Journal of biomedical informatics*, 43(5):762–773.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC'2012)*, pages 2214–2218, Istanbul, Turkey.

Antonio Toral. 2013. Hybrid selection of language model training data using linguistic information and perplexity. In *Proceedings of the second workshop on hybrid approaches to translation*, pages 8–12.

Nicola Ueffing, Gholamreza Haffari, and Anoop Sarkar. 2007. Transductive learning for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 25–32, Prague, Czech Republic. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 6000–6010.

Joern Wuebker, Patrick Simianer, and John DeNero. 2018. Compact personalized models for neural machine translation. *arXiv preprint arXiv:1811.01990*.

Keiji Yasuda, Ruiqiang Zhang, Hirofumi Yamamoto, and Eiichiro Sumita. 2008. Method of selecting training data to build a compact and efficient translation model. In *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-II*.

Jiajun Zhang and Chengqing Zong. 2016. Exploiting source-side monolingual data in neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545, Austin, Texas. Association for Computational Linguistics.

FJWU participation for the WMT20 Biomedical Translation Task

Sumbal Naz[1], **Sadaf Abdul Rauf**[1,2], **Noor e Hira**[1], **Syeda Abida**[1] **and Sami Ul Haq**[3]
[1] Fatima Jinnah Women University, Pakistan
[2] Univ. Paris-Saclay, CNRS, LIMSI France
[3] National University of Sciences and Technology, Pakistan
{sadaf.abdulrauf,sumbalnaz01,noorehira94,sami.haq99}@gmail.com

Abstract

This paper reports system descriptions for FJWU-NRPU team for participation in the WMT20 Biomedical shared translation task. We focused our submission on exploring the effects of adding in-domain corpora extracted from various out-of-domain sources. Systems were built for French to English using in-domain corpora through fine tuning and selective data training. We further explored BERT based models specifically with focus on effect of domain adaptive subword units.

1 Introduction

In this paper, we present Neural Machine Translation (NMT) systems developed by Fatima Jinnah Women University for participation in WMT20, Biomedical shared Translation task. The systems are developed for translating English/French (EN/FR) in both directions for biomedical domain using fairseq (Ott et al., 2019) and BERT (Devlin et al., 2018). To tackle in-domain corpus shortage challenge, selective data training and fine tuning are explored. We focused our submission on investigating the effects of adding in-domain corpora extracted from out-of-domain sources of various domains, objective was to study the effect of domain non-relatedness in schemes involving data selection through information retrieval or any sentence selection method. We further explored BERT based models specifically with focus on effect of domain adaptive subword units.

Neural Machine Translation systems have shown substantial growth with the ongoing introduction of new tool kits and training techniques to support developers in training models (Bahdanau et al., 2014; Wu et al., 2016). But the availability and cleaning of domain related corpora to achieve terminology advantage and fluency is still a challenge for many researchers as the accessible corpora is relatively small in size and comparatively noisy. In order to improve in-domain NMT systems, out-of-domain data is used and the most common method is to fine tune pre trained NMT models on in-domain data and selective data training (Hira et al., 2019).

Our last years submission presented promising results using selective data training incorporating data retrieved from News Commentary corpus by building two layered RNN systems. We extend our framework to study the quality of retrieved sentences from 3 more parallel corpora. We did not restrict to parallel data for mining biomedical sentences, rather this year we included monolingual data in our framework and studied the effect of using Back Translations (BT) in our framework. For building NMT models we explored subword units and report the results on using pre-trained BERT fused embedding.

2 Data Selection Architecture

Improving translation quality is a challenging task especially for domains where enough in-domain parallel corpus is not available to train a good translation system. To overcome data scarcity problem, several data selection techniques have been proposed over the years including information retrieval (IR) (Rauf and Schwenk, 2011), edit distances (Wang et al., 2013), cross entropy measures (Axelrod et al., 2011) and several others. We used the approach of relative query sentences using information retrieval to retrieve matching sentences from general domain corpora.

French-English is not a resource scarce language pair and has numerous parallel corpora available for various domains. There exist sizable corpus for the Biomedical domain to train the initial systems, but the great difference of terminologies and language jargon in various sub domains makes it challenging as the results of previous years bio med-

Proceedings of the 5th Conference on Machine Translation (WMT), pages 849–856

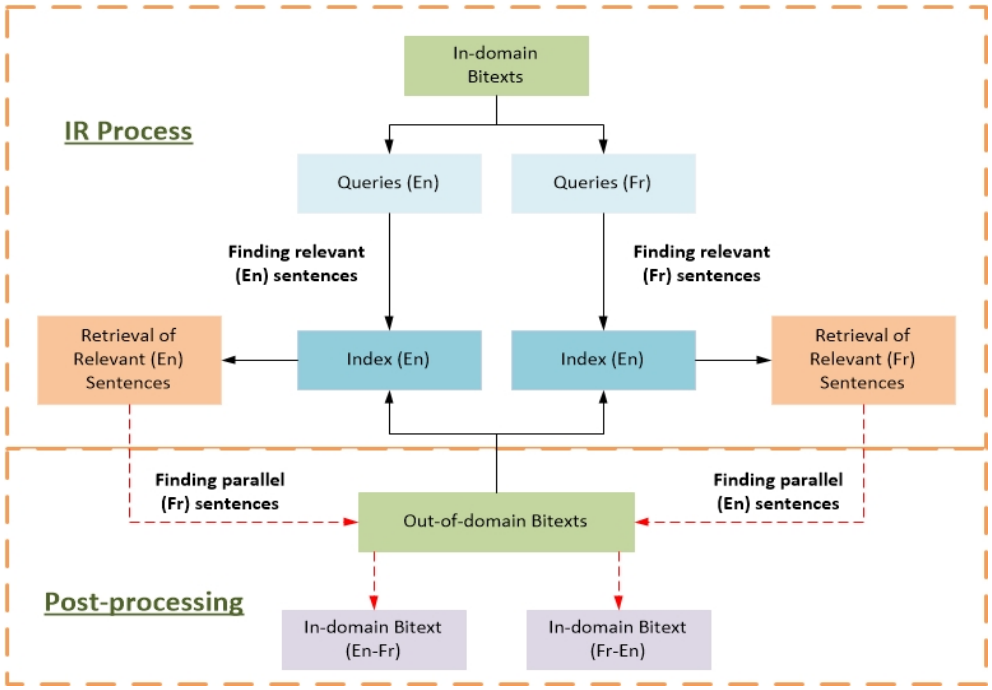


Figure 1: Data selection Architecture.

ical tasks indicate. Parallel corpora extracted from "other" easily available corpora, like comparable corpora and monolingual corpora do help improve MT performance (Abdul-Rauf and Schwenk, 2009; Abdul-Rauf et al., 2016). But, what is the effect of the domain of the corpus used to find the related sentences, is the question we focus on in our data selection design.

Our aim is to study the improvements achieved by using the sentences from different genre/domain of corpora. However, to be able to extract sizeable amount of biomedical sentences, the corpora should not be very unrelated, for example, the Europarl corpus (Koehn, 2005) which is composed of Parliament proceedings would not be a good choice[1]. Our intent was to do a comparative study of quality of extracted sentences from varied but yet not too far off domain corpora. Thus, for mining related sentences from general domain corpora we used Books[2], News Commentary[3] and WikiPedia[4] corpus obtained from Open Parallel Corpus (OPUS) (Tiedemann, 2012).

French WikiPedia[5] (FrWikipediaMono) was also used which was available as monolingual corpus and was translated to English.

Corpus	Corpus Size	Retrieved Sentences	Unique Sentences
Books	127085	1235684	42827
News Commentary	209479	1244026	72011
WikiPedia	818302	1236092	105880
FrWikiPediaMono	8766978	938834	162743

Table 1: Number of sentences retrieved for each corpus for top-2 using French side of Medline tiltles as queries.

Our data selection strategy is graphically presented in figure 1. We followed the data selection approach based on IR as proposed by (Abdul-Rauf et al., 2016). The choice of corpus to use as queries was a critical one: queries should have maximum biomedical terminologies to enable targeting and choosing domain specific sentences from the general domain corpora. We chose Medline titles as queries hypothesising on the fact that the title essentially contains the specific domain terminology. We used English side of Medline titles as queries when retrieving similar sentences

[1] It must have some biomedical sentences from parliamentary debates on health issues, but the amount will be very little.

[2] http://opus.nlpl.eu/Books-v1.php

[3] http://opus.nlpl.eu/News-Commentary-v14.php

[4] http://opus.nlpl.eu/Wikipedia-v1.0.php

[5] https://www.dropbox.com/s/le4yxfijxt0uiia/frwiki-20181001-corpus.xml.bz2?dl=0

from English side of the corpora and French side of Medline titles as queries for IR from French side. We retrieved `10-best` sentences and experimented with `top-1`, `top-2` and `top-3` sentences as shown in section 4. Table 1 shows the number of retrieved sentences per each corpus and the unique sentences chosen from these to build our models.

Corpus	Sentences
In-domain training data	
Ufal	2358164
Scielo	6827
EDP	2200
Medline Abstracts	51520
Medline Titles	567257
Selective IR training data	
News Commentary-IR1	40645
News Commentary-IR2	60671
News Commentary-IR3	75347
Books-IR1	27938
Books-IR2	39901
Books-IR3	48291
WikiPedia-IR1	46439
WikiPedia-IR2	74595
WikiPedia-IR3	97554
Monolingual	
FrWikipediaMono-IR1	81851
FrWikipediaMono-IR2	133259
FrWikipediaMono-IR3	177266
Development data	
Scielo	3606
EDP	295
Khresmoi	1452
Test Data	
Medline 18	231
Medline 19	442

Table 2: Sentence Pairs for Training, Development and Test sets. Sizes are given for cleaned corpora.

3 Corpora

In this section, we present details of corpora used to train our systems, pre-processing and training parameters. We used the in-domain corpora provided by the organizers along with our mined in-domain sentences from the general domain corpora. The in-domain corpora included were:

- Ufal medical corpus, where a subset of medical corpora were extracted including CESTA, ECDC, EMEA, Subtitles and patTR medical corpus. (Yepes et al., 2017)
- Scielo corpus that included scientific biodomain articles. (Neves et al., 2016)
- EDP dataset containing documents from EDP database for scientific publications. (Névéol et al., 2018)
- Medline abstracts and titles from publications.(Bawden et al., 2019)

Books, News Commentary, WikiPedia and FrWikipediaMono corpora were used as the outdomain corpora to perform in data selective training experiments by extracting relevant in-domain sentences as explained in section 2. Development set included EDP, Scielo and Khresmoi (Dušek et al., 2017). Medline test corpora provided by WMT18 (Neves et al., 2018) and WMT19 (Bawden et al., 2019) were used as test sets.

3.1 Pre-processing

Our pre-processing pipeline includes data cleaning, punctuation normalization, tokenization, truecasing and subword segmentation.

Data cleaning was done to remove noisy data. Some of the provided corpora, including EDP, Scielo and Subtitles, were not completely aligned so we used Microsoft's bilingual sentence aligner[6] (Moore, 2002) for their complete alignment. Empty lines, hyperlinks, parenthesis, white spaces if present at the beginning of sentences were removed. Sentences having more than 120 tokens were dropped using Moses cleaning scripts (Koehn et al., 2007), punctuation and normalization was also applied. Table 2 shows our corpus sizes in terms of number of sentences (after cleaning).

For our French to English systems, we tokenized the corpora using Moses tokenizer[7]. Byte Pair Encoding (BPE) sub word units with a vocabulary of 32K units were computed on true cased data using `subword-nmt` (Sennrich et al., 2015) . BertTokenizer[8] was only used for our submitted English to French system.

[6] https://www.microsoft.com/en-us/download/details.aspx?id=52608

[7] https://github.com/moses-smt/mosesdecoder/blob/master/scripts/tokenizer/tokenizer.perl

[8] https://huggingface.co/transformers/model_doc/bert.html

ID	Train Set	Size	Test sets	
French to English		(No of sentences)	Medline18	Medline19
Baseline	WMT	2,985,968	29.6	34.7
S1	WMT + News Commentary-IR1	3,026,613	32.5	35.1
	WMT + News Commentary-IR2	3,046,639	**33.1**	**36.8**
	WMT + News Commentary-IR3	3,061,315	29.5	34.2
S2	WMT + Books-IR1	3,013,906	32.7	36.4
	WMT + Books-IR2	3,025,869	**32.9**	**37.0**
	WMT + Books-IR3	3,034,259	29.4	32.7
S3	WMT + WikiPedia-IR1	3,032,407	**33.1**	36.4
	WMT + WikiPedia-IR2	3,060,563	31.9	**37.2**
	WMT + WikiPedia-IR3	3,083,522	29.4	33.6
S4	WMT + FrWikipediaMono-IR1	3,067,819	32.3	36.1
	WMT + FrWikipediaMono-IR2	3,119,227	**32.5**	**36.9**
	WMT + FrWikipediaMono-IR3	3,163,234	31.4	35.5

Table 3: BLEU scores for (BERT-fused NMT) French to English Models trained with selective data training from out-of-domain corpus

3.2 Training and Parameters

We used Fairseq (Ott et al., 2019), an open-source toolkit for training simple transformer (Vaswani et al., 2017) model and Bert-nmt[9] for training BERT-fused NMT systems. Our experiments can be grouped in three categories depending upon the corpora used during training and their training approach. I) Models trained using all the in-domain corpora provided by WMT. II) Models trained on all the in-domain WMT corpora with addition of in-domain corpus retrieved from out-of-domain corpora using IR. III) Models fine tuned on Medline abstracts and titles (since test corpus is from Medline), from few models built in second category. We used transformer base (Vaswani et al., 2017) architecture provided by fairseq as `transformer_iwslt_de_en`. Adam optimizer and a batch size of 4K words was used in all the experiments. Training was done till complete convergence, models were checked for improvements on test data, and training was stopped if no further improvement in BLEU scores is calculated after 2-3 successive checkpoints. For BERT-fused NMT models same training parameters were used as for NMT models except that multilingual bert base was incorporated during training following the approach of (Zhu et al., 2020)

[9]https://github.com/bert-nmt/bert-nmt

4 Experiments and Results

In this section we report the details of the experiments we performed for our participation in the WMT20 Biomedical task. We performed several different experiments to investigate the performance of NMT with different training approaches. Several different models were trained for French to English translation direction and one model was trained for English to French translation direction. The experiments were conducted as an extension of our last year's submission (Hira et al., 2019) with two different objectives. First, to investigate the performance of BERT-fused NMT over state-of-the art transformer model and the other to explore the effect of out-of-domain corpus used for selective data training. We evaluated our models on Medline 18 and Medline 19 test sets, scores were calculated using sacrebleu (Post, 2018).

4.1 Corpus Selection for Selective Data training

The significant gains in performance due to selective data training, as achieved in our WMT19 participation moved us to explore further to catego-

ID	Approach	Training sets	Test sets	
French to English			**Medline 18**	**Medline 19**
M1	Transformer	WMT	33.2	36.3
M2	BERT-fused transformer (cased)	WMT	29.5	32.6
M3	BERT-fused transformer (uncased)	WMT	29.6	34.7
R1	BERT-fused transformer (SD)	WMT + all IR2	31.8	37.2
R2	R1 fine tuned	WMT + all IR2	35.1	**38.4**
R3	BERT-fused transformer (SD)	WMT + all IR3	29.7	34.0
R4	R3 fine tuned	WMT + all IR3	**47.5**	36.8
English to French				
M4	Transformer	WMT + Books + WikiPedia	**32.5**	**35.8**

Table 4: BLEU scores for BERT-fused NMT with IR incorporated French to English models.

rize which out-of-domain corpus is a better choice. We extended out-of-domain corpora to four different resources for selective data training. Alongwith News Commentary, which was also used in WMT19 participation, we extended the list with WikiPedia corpus, Books corpus and back translated FrWikipediaMono corpus. These were used to build four different sets of models from $S1$ to $S4$ as listed in Table 3. Adding the IR retrieved data has unanimously helped improve the scores to almost 3 BLEU points on both test sets.

These models were trained using WMT20 in-domain corpora with addition of selective `top-1`, `top-2` and `top-3` retrieved IR sentences. $S1$ represents models built using additional News commentary IR corpus. Best scores were obtained on `top-2` yielding 33.1 and 36.8 BLEU points on Medline 18 and Medline 19 test sets. $S2$ consists of models trained on additional Books IR corpus and best scores were again achieved on `top-2` giving 32.9 and 37.0 points on Medline 18 and Medline 19 test sets. $S3$ comprises of models trained using additional WikiPedia IR corpus that reveal change in trend by giving best points 33.1 on `top-1` for Medline 18 and 37.2 on `top-2` for Medline 19 test sets. Similarly, systems represented by $S4$ show the effect of adding back translated FrWikipediaMono IR corpus in training set, that followed the trend of $S1$ and $S2$ giving best points 32.5 and 36.9 on `top-2` for Medline 18 and Medline 19 respectively. We can safely conclude that `top-2` IR retrieved sentences give us the best score. As for the effect of domain/type of the corpus used for IR, we don't see any significant advantage of any corpus over the other. For example, News Commentary and Books are very different corpora, but still sentences from both the corpora yield more or less the same improvement. Same is the case with WikiPedia, whether parallel or monolingual. This is an expected outcome as the IR process retrieves the sentences most relevant to the query sentence (Medline titles in our case).

4.2 BERT-fused NMT

To target our second objective, investigation of BERT-fused NMT performance over transformer model, we trained three models using in-domain data provided by WMT20 Bio-medical translation task; $M1$, $M2$ and $M3$. And four models, $R1$ to $R4$, using additional IR data, for French to English translation direction. Whereas 1 model ($M4$) for English to French translation direction, as shown in table 4. $M1$ was trained with simple transformer architecture without BERT fusion and it scored 33.2 and 36.3 BLEU points on Medline 18 and Medline 19 test sets respectively. $M2$ was trained under BERT-fused NMT setting with cased multilingual BERT base fused in transformer architecture. This model yielded 29.5 BLEU score on Medline 18 and 32.6 BLEU score on Medline 19 test set. Unexpectedly $M2$, despite being trained in BERT-fused NMT setting, didn't show improvements in BLEU points over simple transformer model ($M1$). One reason of this unexpected decrease in the BLEU scores of $M2$ over $M1$ could be the use of BERT trained on general domain. It seems that BERT

trained on much huge general domain corpus has suppressed the learned parameters from in-domain training corpus. $M3$ was trained as similar to $M2$ but with uncased BERT, to explore the difference in the performance of cased and uncased BERT model, and it showed little improvement than $M2$ on Medline 18 test data with a difference of only 0.1 BLEU points whereas an increase of 2.1 BLEU points on Medline 19 test set, as listed in Table 4. Based on this result, we selected uncased BERT model for our further experiments.

Further, we tried to evaluate the performance of selective data training in BERT-fused NMT setting, and trained four models for this investigation as shown in Table 4. $R1$ was trained over in-domain WMT20 corpus concatenated with `top-2` queried all IR data, since these proved to be most beneficial as shown by the results from section 4.1. $R1$ scored 31.8 and 37.2 BLEU points on Medline 18 and Medline 19 test sets respectively. Comparing $R1$ with $M2$ depicts that BERT-fused NMT also benefits from data selective training approach, as the results show considerable increase in BLEU points, increasing 2.3 and 2.5 BLEU scores on Medline 18 and Medline 19 respectively by adding only 0.3M (308426 sentences) IR data. Though the addition of IR data for training $R1$ improved scores compared to $M2$ but did not outperform $M1$ which initiated the need to verify our assumption that general domain BERT is suppressing the learned parameters from in-domain training data. So, for verification we fine tuned $R1$ on Medline abstracts and Medline titles data to train a new system $R2$. $R2$ showed improvements in scores as fine tuned on in-domain corpus (Medline abstracts and titles). It gave highest BLEU score points of 38.4 for medline 19 test set and also producing 35.1 BLEU points on Medline 18. This verify that our assumption about the unexpected results of BERT-fused NMT model was correct. Another model $R3$ was built to test the effect of queried IR data. $R3$ was trained over in-domain WMT20 corpus concatenated with `top-3` queried all IR data. It yielded 29.7 and 34.0 BLEU scores on Medline 18 and Medline 19 respectively. $R3$ is then fine tuned on Medline abstracts and Medline titles data to train a new system $R4$. $R4$ scored highest BLEU points of 47.5 for Medline 18 and gave 36.8 BLEU points for Medline 19 test set.

For English to French translation direction, we trained transformer model with hugging face BERT tokenizer instead of BERT-fused NMT ($M4$). The model was trained with transformer architecture on in-domain data and selective data from Books and WikiPedia corpus. This model ranked third in official results provided by WMT20 and scored 32.5 and 35.8 BLEU points on Medline 18 and Medline 19 test sets respectively.

5 Related Work

Numerous challenges arise when dealing with biomedical data used for translation due to limited size of corpus and unstructured alignments. Various approaches have been adopted by researchers in WMT biomedical translation. (Khan et al., 2018) submitted a NMT system that combined in-domain data set and used transfer learning approach to train the model along with ensemble learning. (Huck et al., 2018) trained by using transformer architecture using biomedical and news domain and employed cascaded word segmentation along with BPE. (Tubay and Costa-jussà, 2018) emphasize on using multi-source approach like Romance languages with in-domain data by implementing transformer architecture using OpenNMT in PyTorch. (Carrino et al., 2019) created terminology list for biomedical words using BabelNet API, inserted the information at a token level and trained NMT system using transformer model (Vaswani et al., 2017). (Hira et al., 2019) used selective learning for building additional corpus from out-of-domain data and incorporated transfer learning approach by using recurrent encoder decoder NN model for training of in-domain biomedical data.(Peng et al., 2019) trained their Transformer model on in-domain and out-of-domain data for six translations using transfer learning methods. The model used attention mechanism along with RELU activation function yielding better results for in-domain biomedical data. (Saunders et al., 2019) used transfer learning using Bayesian Interpolation for multi-domain data for ensemble weighting. (Soares and Krallinger, 2019) participated in WMT19 with four translation directions by creating concatenating corpora from UMLS, out-of-domain and in-domain data and trained the systems using Transformer model.

6 Conclusion

In this paper, we present our submission for WMT20 Biomedical tasks. Our model trained for English to French language direction ranked third in official scores provided by WMT20. We trained

different models to investigate the performance of BERT-fused NMT over transformer model and to explore the effect of selective data training in BERT-fused NMT for French to English language direction. Results show decline in performance of BERT-fused NMT models over transformer architecture as general domain BERT suppressed the learned parameters from in-domain training corpus. BERT-fused models yielded better results when fine tuned on in-domain corpus and trained with IR data.

Acknowledgments

This study is funded by the National Research Program for Universities (NRPU) by Higher Education Commission of Pakistan (5469/Punjab/NRPU/R&D/HEC/2016).

References

Sadaf Abdul-Rauf and Holger Schwenk. 2009. On the use of comparable corpora to improve SMT performance. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 16–23, Athens, Greece. Association for Computational Linguistics.

Sadaf Abdul-Rauf, Holger Schwenk, Patrik Lambert, and Mohammad Nawaz. 2016. Empirical use of information retrieval to build synthetic data for smt domain adaptation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(4):745–754.

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. 2011. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 355–362.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Rachel Bawden, Kevin Bretonnel Cohen, Cristian Grozea, Antonio Jimeno Yepes, Madeleine Kittner, Martin Krallinger, Nancy Mah, Aurelie Neveol, Mariana Neves, Felipe Soares, et al. 2019. Findings of the wmt 2019 biomedical translation shared task: Evaluation for medline abstracts and biomedical terminologies. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 29–53.

Casimiro Pio Carrino, Bardia Rafieian, Marta R. Costa-jussà, and Josà© A. R. Fonollosa. 2019. Terminology-aware segmentation and domain feature for the wmt19 biomedical translation task. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 153–157, Florence, Italy. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Ondřej Dušek, Jan Hajič, Jaroslava Hlaváčová, Jindřich Libovický, Pavel Pecina, Aleš Tamchyna, and Zdeňka Urešová. 2017. Khresmoi summary translation test data 2.0. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Noor-e Hira, Sadaf Abdul Rauf, Kiran Kiani, Ammara Zafar, and Raheel Nawaz. 2019. Exploring transfer learning and domain data selection for the biomedical translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 158–165, Florence, Italy. Association for Computational Linguistics.

Matthias Huck, Dario Stojanovski, Viktor Hangya, and Alexander Fraser. 2018. Lmu munichâ™s neural machine translation systems at wmt 2018. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 659–665, Belgium, Brussels. Association for Computational Linguistics.

Abdul Khan, Subhadarshi Panda, Jia Xu, and Lampros Flokas. 2018. Hunter nmt system for wmt18 biomedical translation task: Transfer learning in neural machine translation. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 666–672, Belgium, Brussels. Association for Computational Linguistics.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86. Citeseer.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*, pages 177–180. Association for Computational Linguistics.

Robert C Moore. 2002. Fast and accurate sentence alignment of bilingual corpora. In *Conference of the Association for Machine Translation in the Americas*, pages 135–144. Springer.

Aurélie Névéol, Antonio Jimeno Yepes, L Neves, and Karin Verspoor. 2018. Parallel corpora for the biomedical domain.

Mariana Neves, Antonio Jimeno Yepes, and Aurélie Névéol. 2016. The scielo corpus: a parallel corpus of scientific publications for biomedicine. In

Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16), pages 2942–2948.

Mariana Neves, Antonio Jimeno Yepes, Aurélie Névéol, Cristian Grozea, Amy Siu, Madeleine Kittner, and Karin Verspoor. 2018. Findings of the wmt 2018 biomedical translation shared task: Evaluation on medline test sets. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 324–339.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. *arXiv preprint arXiv:1904.01038*.

Wei Peng, Jianfeng Liu, Liangyou Li, and Qun Liu. 2019. Huawei's nmt systems for the wmt 2019 biomedical translation task. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 166–170, Florence, Italy. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting bleu scores. *arXiv preprint arXiv:1804.08771*.

Sadaf Abdul Rauf and Holger Schwenk. 2011. Parallel sentence generation from comparable corpora for improved smt. *Machine translation*, 25(4):341–375.

Danielle Saunders, Felix Stahlberg, and Bill Byrne. 2019. Ucam biomedical translation at wmt19: Transfer learning multi-domain ensembles. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 171–176, Florence, Italy. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Felipe Soares and Martin Krallinger. 2019. Bsc participation in the wmt translation of biomedical abstracts. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 177–180, Florence, Italy. Association for Computational Linguistics.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *Lrec*, volume 2012, pages 2214–2218.

Brian Tubay and Marta R. Costa-jussà. 2018. Neural machine translation with the transformer and multi-source romance languages for the biomedical wmt 2018 task. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 678–681, Belgium, Brussels. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Longyue Wang, Derek F Wong, Lidia S Chao, Junwen Xing, Yi Lu, and Isabel Trancoso. 2013. Edit distance: A new data selection criterion for domain adaptation in smt. In *Proceedings of the International Conference Recent Advances in Natural Language Processing RANLP 2013*, pages 727–732.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Antonio Jimeno Yepes, Aurélie Névéol, Mariana Neves, Karin Verspoor, Ondřej Bojar, Arthur Boyer, Cristian Grozea, Barry Haddow, Madeleine Kittner, Yvonne Lichtblau, et al. 2017. Findings of the wmt 2017 biomedical translation shared task. In *Proceedings of the Second Conference on Machine Translation*, pages 234–247.

Jinhua Zhu, Yingce Xia, Lijun Wu, Di He, Tao Qin, Wengang Zhou, Houqiang Li, and Tie-Yan Liu. 2020. Incorporating bert into neural machine translation. *arXiv preprint arXiv:2002.06823*.

Huawei's Submissions to the WMT20 Biomedical Translation Task

Wei Peng[1], Jianfeng Liu[1], Minghan Wang[2], Liangyou Li[3], Xupeng Meng[1], Hao Yang[2], Qun Liu[3]
[1]Artificial Intelligence Application Research Center, Huawei Technologies
{peng.wei1,liujianfeng,mengxupeng}@huawei.com
[2]Huawei Translation Service Center, Huawei Technologies
{wangminghan,yanghao30}@huawei.com
[3]Noah's Ark Lab, Huawei Technologies
{liliangyou,qun.liu}@huawei.com

Abstract

This paper describes Huawei's submissions to the WMT20 biomedical translation shared task. Apart from experimenting with fine-tuning on domain-specific bitexts, we explore effects of in-domain dictionaries on enhancing cross-domain neural machine translation performance. We utilize a transfer learning strategy through pre-trained machine translation models and extensive scope of engineering endeavors. Four of our ten submissions achieve state-of-the-art performance according to the official automatic evaluation results, namely translation directions on English⇔French, English→German and English→Italian.

1 Introduction

Neural machine translation (NMT) models built upon the Transformer architecture (Vaswani et al., 2017) start to dominate the leader board of WMT biomedical shared tasks in recent years (Bawden et al., 2019). In-domain data (parallel and monolingual corpora) have been widely used in finetuning general domain NMT models. Despite ongoing improvements on the translation quality observed from recent biomedical shared tasks, domain adaptation remains an open problem. The in-domain data is hard to obtain and, as a consequence, greatly limits the cross-domain translation capability an NMT model can offer. Domain terminologies, on the other hand, are regarded as critical resources to improve the quality of machine translation by mitigating effects of scarce in-domain bitexts (Bawden et al., 2019). However, few research works leverage domain-specific terminologies (or dictionaries) in training cross-domain NMT systems.

In this paper, we present the system architecture and research approaches underpinning Huawei's submissions to the WMT20 biomedical translation task. We implement two NMT systems to maximize the performances of the shared task. The system I is an in-house NMT system built upon the transformer-big architecture (Vaswani et al., 2017) and trained using general domain data. We explore means to enhance cross-domain coverage of an NMT model by finetuning the NMT model with in-domain bitexts. We also investigate the effects of domain dictionaries in this domain adaptation process. Reusing pre-trained models has been regarded as an efficient way of transfer learning. Pre-trained NMT models (Ng et al., 2019) are adopted in the system II to this end.

All NMT systems are evaluated against the test set released in the WMT19 biomedical shared task. We submitted translated results for a total of ten language directions between English (EN) and other five languages including French (FR), German (DE), Italian (IT), Russian (RU) and Chinese (ZH). Four of the submissions achieve the best BLEU scores according to the official automatic evaluation results. Substantial increases in BLEU scores are recorded in translation directions of DE→EN (+3.9 BLEU), ZH→EN (+3.5 BLEU), and EN→DE (+2.8 BLEU) compared to our submissions last year (Peng et al., 2019). The improvements on EN⇔DE can be ascribed to strong pre-trained NMT baseline models and a series of optimization techniques, for example, in-domain data augmentation and a reranking method with strong language models. High-quality in-domain data and large-scale back-translation contribute to the improvements of the ZH→EN model.

2 The Data

Table 1 captures the number of sentences pairs used in this shared task. The system I is trained using in-house general domain data (OOD) and finetuned on the in-domain data (IND) provided by

Proceedings of the 5th Conference on Machine Translation (WMT), pages 857–861
Online, November 19–20, 2020.

Directions		Train				Dev.	Test	Vocab.
		OOD	IND	IND-Dict.	IND-Aug.			
System I	EN→FR	146M	4M	59K	-	4K	440	40K
	FR→EN	186M	4M	59K	-	4K	417	40K
	EN→IT	83M	219K	-	-	3.8K	400	40K
	IT→EN	150M	219K	-	-	3.8K	400	40K
	EN→ZH	164M	-	59K	-	5K	448	50K
	ZH→EN	200M	-	59K	55M	5K	115	50K
System II	EN→DE	-	40K	-	56K	435	-	42K
	DE→EN	-	40K	-	56K	373	-	42K
	EN→RU	-	54K	-	-	300	-	32K(EN)/31K(RU)
	RU→EN	-	54K	-	-	300	-	32K(EN)/31K(RU)

Table 1: Data used for training and finetuning systems I and II. Note that "IND-Dict." refers to the in-domain dictionary. "IND-Aug." is the augmented data derived from processing IND data. For the system I, "IND-Aug." is created from back-translating monolingual data. For the system II, "IND-Aug." is the pre-processed IND data in combination with the data selected from some OOD data based on the similarity to the Medline data. M is for "million," and K stands for "thousand".

WMT20.[1] The in-domain data consist of bitexts from EMEA (Tiedemann, 2012), UFAL,[2] Pubmed, and Medline.[3] The data is processed by methods in the next section. The test data for the system I are from the WMT19 shared task.

The data used for finetuning the system II are different from those for the system I. The system II only focuses on Medline as we discovered it is the most effective IND data for this shared task. The development (dev.) set for the system II is the OK-aligned test data from the WMT19 biomedical shared task.

A batch of monolingual Medline data in English dated before July 2018 has been extracted to provide a basis for data augmentation and noisy channel model reranking (Ng et al., 2019). It produces the augmented IND data for the ZH→EN translation direction via back-translation ("IND-Aug." in Table 1). Due to time and resource constraints, we could not fully explore this monolingual Medline data in other translation directions.

3 The Approaches

The proposed systems are finetuned and enhanced using the following methods. All models are trained on Tesla V100 GPUs. Systems I and II use batch sizes of 6,144 and 8,000 tokens respectively in the finetuning process.

3.1 In-domain Dictionary

Bilingual dictionaries have been studied in the machine translation community for various purposes. The lexicons are used to enhance the translation quality for rare and unknown words in the parallel corpus (Zhang and Zong, 2016). Research works in domain adaptation for NMT showed that incorporating domain-specific dictionaries is a viable solution (Hu et al., 2019; Thompson et al., 2019; Peng et al., 2020). Inspired by these studies, we apply domain-specific dictionaries derived from SNOMED-CT, [4] which is a collection of multilingual clinical terminology, to finetune general domain NMT models to boost cross-domain coverage. The dictionaries are treated as bitexts attached to the end of training data.

3.2 Reranking

Apart from adopting a data-driven approach mentioned above, we also apply a transfer learning approach by reusing the publicly available pre-trained NMT models provided at fairseq (Ott et al., 2019). [5] After finetuning the selected pre-trained NMT models on the in-domain data, we apply a noisy channel model reranking method (Ng et al., 2019). The weights λ in Equation 1 are learned with a

[1] http://www.statmt.org/wmt20/biomedical-translation-task.html

[2] https://ufal.mff.cuni.cz/ufal_medical_corpus

[3] https://github.com/biomedical-translation-corpora/corpora

[4] https://www.nlm.nih.gov/healthit/snomedct/index.html

[5] https://github.com/pytorch/fairseq

System I	EN→FR	FR→EN	EN→IT	IT→EN	EN→ZH	ZH→EN
baseline	38.98	38.31	30.85	35.73	36.22	34.37
+ ft BS, IND	-	-	31.04	35.93	-	-
+ ft IND, IND-Dict.	41.66	38.44	-	-	-	-
+ ft BS,IND-Dict.,IND-Aug.	-	-	-	-	35.90	35.66
WMT19 Submission	42.41	38.24	-	-	37.09	32.16
WMT20 Submission	**43.51**	**44.45**	**42.57**	**49.74**	**45.46**	**35.28**
WMT20 Best Official	**43.51**	**44.45**	**42.57**	**50.11**	**46.86**	**35.28**

Table 2: BLEU scores of the system I on all related submissions. The baseline models are finetuned (ft) in various configurations, including mixed finetuning on in-house OOD data (aka BS), IND bitexts, "IND-Dict." and the augmented IND data ("IND-Aug."). Note that the WMT20 best official score for ZH→EN excludes those results currently under investigation.

random search for the best performing candidate on the validation data.

$$\lambda_1 logP(y|x) + \lambda_2 logP(y) + \lambda_3 logP(x|y) \quad (1)$$

Due to time constraints, we did not implement the reranking approach on the system I.

3.3 Data Processing

A data processing pipeline is applied to enhance the quality of training data:

- Data cleaning is implemented to filter out noisy data. An important step is to handle misalignment in the parallel corpus. An alignment model trained by fast-align (Dyer et al., 2013) [6] is applied to this end (Lu et al., 2018). In addition, we remove bitexts with a source and target sentence length ratio exceeding a certain threshold (i.e., 2.5). A language detection tool [7] is used to filter out bitexts with abnormal language patterns, i.e., sentences with undesirable *langid*. Other noisy data, such as those with HTML tags and extra spaces, are removed.
- Scripts from Moses (Koehn et al., 2007) are used to perform punctuation normalization and tokenization. SentencePiece (Kudo and Richardson, 2018) segments words into subwords.
- We extract "in-domain" data which are close to Medline from general domain data by using TFIDF-based similarities. Similar data augmentation approaches can be identified in Wang et al. (2017) and Peng et al. (2020).
- Post-processing is performed after decoding to detokenize subwords and remove undesirable spaces between special characters and numbers, i.e., converting "23 - 25" into "23-25".

[6] https://github.com/clab/fast_align
[7] https://github.com/aboSamoor/polyglot

4 Experimental Results

The systems are trained with OOD data and finetuned using IND data to produce the submitted results. We benchmarked the submissions using WMT19 test data. The BLEU scores are calculated using the MTEVAL script from Moses (Koehn et al., 2007). Results are shown in Table 2 and Table 4. The final two rows demonstrate the scores of our submissions on this year's test sets and the best official records released by the organizers.

4.1 English ⇔ French

The system I is our in-house system equipped with an extensive data processing pipeline to handle noisy data, i.e., the application of sentence alignment and language detection tools. Our EN→FR and FR→EN submissions achieve the best official results in the WMT20 shared task. IND bitexts and "IND-Dict." have contributed to up to 2.7 BLEU in enhancing the baseline performance. We presume the improvement is due to the enhanced domain coverage the IND data brought forth. Note that even with much larger OOD bitexts than last year, the system produces similar benchmark scores. It appears an over-representation of OOD data is not helpful in cross-domain NMT. An analysis of domain coverage is performed to investigate the effect of IND information on cross-domain translation. We count the number of unique terms (1-2 grams)

Data	EN→FR		FR→EN	
	Unigrams	Bigrams	Unigrams	Bigrams
OOD	2,763	5,752	2,989	6,317
OOD + IND + IND-Dict.	2,773 (+10)	5,827 (+75)	2,997 (+8)	6,372 (+55)

Table 3: Domain coverage analysis for data used to train English⇔French.

System II	EN→DE	DE→EN	EN→RU	RU→EN
baseline	34.12	37.39	-	-
+ ft All Medline	35.58 (+1.46)	39.06 (+1.67)	-	-
+ ft Pre-proc. Medline	36.90 (+1.32)	40.98 (+1.92)	27.30	33.38
+ ft IND-Aug.	37.13 (+0.23)	41.79 (+0.81)	-	-
+ reranking	38.17 (+1.04)	42.74 (+0.95)	-	-
WMT19 Submission	35.39	38.84	-	-
WMT20 Submission	**36.89**	**41.46**	**34.64**	**43.03**
WMT20 Best Official	**36.89**	**41.65**	**39.36**	**43.31**

Table 4: BLEU scores of system II on English ⇔ German. "Pre-proc." stands for "pre-processed." Note that "IND-Aug." contains the pre-processed Medline data and the data derived from OOD via TFIDF selection. Numbers in the brackets depict the incremental increase from the baseline models.

at the intersection of a data source (i.e., the OOD training data) and the test data. Table 3 indicates that the increase of BLEU may be associated with a level of domain coverage enhancement. An increasing number of distinctive IND terms is recorded.

4.2 English ⇔ German

We perform ablation tests on pre-trained NMT models (the system II) in English ⇔ German under various conditions. As shown in Table 4, an EN→DE model finetuned on a preprocessed version of Medline outperforms that trained on the full version of Medline by 1.32 BLEU, indicating the effectiveness of the data preprocessing method. The EN→DE model finetuned on the "IND-Aug." data adds 0.23 to the BLEU score. The performance of the model can be boosted by 1.04 BLEU using the reranking method. Both EN→DE and DE→EN models outperform our last year's submissions significantly by 2.78 and 3.90 BLEU, respectively.

4.3 Other Translation Directions

The submissions for other translation directions are illustrated in Table 2 and Table 3. Note that we did not perform the experiments on the same level as those for English⇔German due to time constraints. It is observed that finetuning on IND data has contributed to improving the performance of baseline models in EN→IT, IT→EN, and ZH→EN directions. The result for EN→ZH is inconclusive, most likely due to potential issues during training.

5 Conclusion

This paper depicts Huawei's submissions to the WMT20 biomedical shared task. For all ten translation directions, we have explored the effects of using IND bitexts and dictionaries on enhancing the performances of cross-domain NMT. We have demonstrated the benefits of the transfer learning strategy of reusing pre-trained NMT models. Four of our ten submissions achieve the best records according to the released WMT20 official results.

Acknowledgments

We would like to show our gratitude to colleagues from Huawei Noah's Ark Lab, AARC, Huawei Translation Service Center, and HTRDC AIE for their support during this work.

References

Rachel Bawden, Kevin Bretonnel Cohen, Cristian Grozea, Antonio Jimeno Yepes, Madeleine Kittner, Martin Krallinger, Nancy Mah, Aurelie Neveol, Mariana Neves, Felipe Soares, Amy Siu, Karin Verspoor, and Maika Vicente Navarro. 2019. Findings of the WMT 2019 biomedical translation shared task: Evaluation for MEDLINE abstracts and biomedical terminologies. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared*

Task Papers, Day 2), pages 29–53, Florence, Italy. Association for Computational Linguistics.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648, Atlanta, Georgia. Association for Computational Linguistics.

Junjie Hu, Mengzhou Xia, Graham Neubig, and Jaime Carbonell. 2019. Domain adaptation of neural machine translation by lexicon induction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2989–3001, Florence, Italy. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Jun Lu, Xiaoyu Lv, Yangbin Shi, and Boxing Chen. 2018. Alibaba submission to the WMT18 parallel corpus filtering task. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 917–922, Belgium, Brussels. Association for Computational Linguistics.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook FAIR's WMT19 news translation task submission. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 314–319, Florence, Italy. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Wei Peng, Chongxuan Huang, Tianhao Li, Yun Chen, and Qun Liu. 2020. Dictionary-based data augmentation for cross-domain neural machine translation.

Wei Peng, Jianfeng Liu, Liangyou Li, and Qun Liu. 2019. Huawei's NMT systems for the WMT 2019 biomedical translation task. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 164–168, Florence, Italy. Association for Computational Linguistics.

Brian Thompson, Rebecca Knowles, Xuan Zhang, Huda Khayrallah, Kevin Duh, and Philipp Koehn. 2019. HABLex: Human annotated bilingual lexicons for experiments in machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1382–1387, Hong Kong, China. Association for Computational Linguistics.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2214–2218, Istanbul, Turkey. European Language Resources Association (ELRA).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Rui Wang, Andrew Finch, Masao Utiyama, and Eiichiro Sumita. 2017. Sentence embedding for neural machine translation domain adaptation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 560–566, Vancouver, Canada. Association for Computational Linguistics.

Jiajun Zhang and Chengqing Zong. 2016. Bridging neural machine translation and bilingual dictionaries. *CoRR*, abs/1610.07272.

Addressing Exposure Bias With Document Minimum Risk Training: Cambridge at the WMT20 Biomedical Translation Task

Danielle Saunders and **Bill Byrne**

Department of Engineering, University of Cambridge, UK

Abstract

The 2020 WMT Biomedical translation task evaluated Medline abstract translations. This is a small-domain translation task, meaning limited relevant training data with very distinct style and vocabulary. Models trained on such data are susceptible to exposure bias effects, particularly when training sentence pairs are imperfect translations of each other. This can result in poor behaviour during inference if the model learns to neglect the source sentence.

The UNICAM entry addresses this problem during fine-tuning using a robust variant on Minimum Risk Training. We contrast this approach with data-filtering to remove 'problem' training examples. Under MRT fine-tuning we obtain good results for both directions of English-German and English-Spanish biomedical translation. In particular we achieve the best English-to-Spanish translation result and second-best Spanish-to-English result, despite using only single models with no ensembling.

1 Introduction

Neural Machine Translation (NMT) in the biomedical domain presents challenges in addition to general domain translation. Text often contains specialist vocabulary and follows specific stylistic conventions. For this task fine-tuning generic pre-trained models on smaller amounts of biomedical-specific data can lead to strong performance, as we found in our 2019 biomedical submission (Saunders et al., 2019). For our WMT 2020 submission we start with strong single models from that 2019 submission and fine-tune them exclusively on the small Medline abstracts training sets (Bawden et al., 2019). This allows fast training on very relevant training data, since the test set is also made up of Medline abstracts.

However, fine-tuning on relevant but small corpora has pitfalls. The small number of training examples exacerbates the effect of any noisy or poorly aligned sentence pairs. We treat this as a form of exposure bias, in that model overconfidence in training data results in poor translation hypotheses at test time.

Our contributions in this system paper are:

- A discussion of exposure bias in the form of imperfect training data, focusing on the biomedical domain.
- An exploration of straightforward ways to mitigate exposure bias via data preparation and training objective.
- A discussion of our 2020 Biomedical task results for single models fine-tuned on small, domain-specific data sets.

1.1 Exposure bias in the biomedical domain

Exposure bias for an autoregressive sequence decoder refers to a discrepancy between decoder conditioning during training and inference (Bengio et al., 2015; Ranzato et al., 2016). During training the decoder generates a hypothesis for the t^{th} output token $\hat{y}_t$ conditioned on $y_{1:t-1}$, the gold target sequence prefix. During inference, the gold target y is unavailable, and $\hat{y}_t$ is conditioned instead on the hypothesis prefix $\hat{y}_{1:t-1}$.

Previous work has interpreted the risk of exposure bias primarily in terms of the model over-relying on correct gold target translations, resulting in error propagation when mistakes are made during inference. We take a different view, focusing on mistakes in the training data which harm the model through teacher-forcing exposure and cause it to make related mistakes during inference.

We identify a specific feature of the Medline abstract training data which caused noticeable translation errors. The data contains instances in which either the source or target sentence contains the

Proceedings of the 5th Conference on Machine Translation (WMT), pages 862–869
Online, November 19–20, 2020.

English source Human translation	[Associations of work-related strain with subjective sleep quality and individual daytime sleepiness]. [Zusammenhang von arbeitsbezogenen psychischen Beanspruchungsfolgen mit subjektiver Schlafqualität und individueller Tagesschläfrigkeit.]
MLE MRT	Zusammenfassung. [Assoziationen arbeitsbedingter Belastung mit subjektiver Schlafqualität und individueller Tagesschläfrigkeit].
English source Human translation	[Effectiveness of Upper Body Compression Garments Under Competitive Conditions: A Randomised Crossover Study with Elite Canoeists with an Additional Case Study]. [Effektivität von Oberkörperkompressionsbekleidung unter Wettkampfbedingungen: eine randomisierte Crossover-Studie an Elite-Kanusportlern mit einer zusätzlichen Einzelfallanalyse.]
MLE MRT	Eine randomisierte Crossover-Studie mit Elite-Kanuten mit einer Additional Case Study wurde durchgeführt. Eine randomisierte Crossover-Studie mit Elite-Kanüsten mit einer Additional Case Study hat zur Wirksamkeit von Oberkörperkompressionsbekleidung unter kompetitiven Bedingungen geführt.

Table 1: Two sentence from the English-German 2020 test set with hypothesis translations from various models, demonstrating the effects of exposure bias from training on imperfectly aligned training sentences. The first MLE example output is completely unrelated to the source sentence, but the second MLE translation is more misleading.

correct translation of the other sentence, but adds information that is not found in translation. For example, the following sentence appears in the English side of en-de Medline abstract training data:

[The effects of Omega-3 fatty acids in clinical medicine]. Effects of Omega-3 fatty acids (n-3 FA) in particular on the development of cardiovascular disease (CVD) are of major interest.

Its corresponding German sentence is

Der Nutzen von Omega-3-Fettsäuren (n-3-FS) in der Medizin, hauptsächlich in der Prävention kardio- und zerebrovaskulärer Erkrankungen, wird aktuell intensiv diskutiert. (Translated: 'The uses of Omega-3 fatty acids in medicine, especially in prevention of cardiovascular and cerebrovascular diseases, are currently heavily discussed.')

Some of the English sentence is present in the German translation, but the square-bracketed article title is not. In this example it might be possible to remove only the segment in square brackets, but in other examples there is even less overlap, while source and target sentences may still be related and therefore challenging to filter. For example, the following English and German sentences also correspond with still less overlap:

[Conflict of interest with industry–a survey of nurses in the field of wound care in Germany , Australia and Switzerland]. Background.

Hintergrund: Pflegende werden zunehmend von der Industrie umworben. (Translated: 'Background: Nurses are being increasingly courted by industry.')

These examples are quite frequent in Medline abstract data, especially in the form of titles. It is common to insert the English title of a non-English article into its translation, marked with square brackets (Patrias and Wendling, 2007). The marked title is not present in the original article. Consequently models trained on English source sentences with titles can behave erratically when given sentences with square-bracketed titles at test time: an exposure bias effect.

One possible approach to this problem is aggressively filtering sentences which may be poorly aligned. However, with such a small training set, this risks losing valuable examples of domain-specific source and target language. We hypothesise that such filtering is not the only way to reduce the effects during inference. Instead, we propose an approach in terms of the parameter fine-tuning scheme with Minimum Risk Training (MRT). Wang and Sennrich (2020) have recently shown MRT as effective for combating exposure bias in the context of domain shift – test sentences which are very different from the training data. We propose that MRT is also more robust against exposure to misaligned training data.

The examples in Table 1 show the different behaviour of MLE and MRT in such cases. In the first example, the MLE hypothesis is unrelated to the source sentence, while the MRT output is relevant. In the second example, the MLE output is more plausible and therefore misleading, as it still misses the first clause which the MRT hypothesis covers. Both MLE and MRT hypotheses are phrased like opening sentences rather than titles, and both feature the untranslated phrase 'Additional Case Study': while MRT may be more robust, it is not immune to exposure bias.

We note that title translations may not exist in the human reference. In these cases failure

to translate the title will not negatively impact BLEU. However, we argue a biomedical translation model should be able to translate such sentences if required. It is also important to note that title translations are not the only case of inexact training pairs, but are simply easily identifiable.

1.2 Document MRT

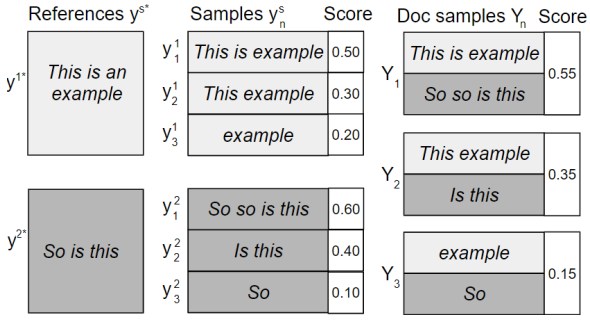


Figure 1: Two MRT schemes with an $S = 2$ sentence minibatch and $N = 3$ samples / sentence. In standard MRT (middle) each sample has a score, e.g. sBLEU. For doc-MRT (right) samples are sorted into minibatch-level 'documents', each with a combined score, e.g. document BLEU. Doc-MRT scores are less sensitive to individual samples, increasing robustness.

Minimum Risk Training (MRT) aims to minimize the expected cost between N sampled target sequences $\boldsymbol{y_n^{(s)}}$ and the corresponding gold reference sequence $\boldsymbol{y}^{(s)*}$ for the S sentence pairs in each minibatch. For translation MRT is usually applied using a sentence-level BLEU (sBLEU) score corresponding to cost function $1 - \text{sBLEU}$, and sentence samples are generated by autoregressive sampling with temperature τ during training (Shen et al., 2016). Hyperparameter α controls sharpness of the distribution over samples. While MRT permits training from scratch, in practice it is exclusively used to fine-tune models.

Doc-MRT is a recently proposed MRT variant which changes sentence cost function to a document cost function, $D(.)$ (Saunders et al., 2020). D measures costs between minibatch-level 'documents' Y^* and Y_n. Y^* is formed of all S reference sentences in the minibatch, and Y_n is one of N sample 'documents' each formed of one sample from each sentence pair $(\boldsymbol{x}^{(s)}, \boldsymbol{y}^{(s)*})$. This permits MRT under document-level scores like BLEU, instead of sBLEU. The n^{th} sample for the s^{th} sentence in the minibatch-level document, $\boldsymbol{y}_n^{(s)}$, contributes the following term to the overall gradient:

$$\frac{\alpha}{N} \sum_{Y:\boldsymbol{y}^{(s)}=\boldsymbol{y}_n^{(s)}} D(Y, Y^*) \nabla_\theta \log P(\boldsymbol{y}_n^{(s)} | \boldsymbol{x}^{(s)}; \theta)$$

In other words the gradient of each sample is weighted by the aggregated document-level scores for documents in which the sample appears.

Figure 1 gives a toy example of doc-MRT scoring samples in context. Document-level metrics aggregate scores across sentence samples, meaning a minibatch with some good samples and some poor samples will not have extreme score variation. Doc-MRT is therefore less sensitive than standard MRT to variation in individual samples.

Doc-MRT has been shown to give better performance than standard MRT for small datasets with a risk of over-fitting, as well as improved robustness to small N. More discussion of these results and a derivation of the document-level loss function can be found in Saunders et al. (2020). Since we are attempting fine-tuning on small datasets and since N is a limiting factor for MRT on memory-intensive large models, the biomedical task is an appropriate application for doc-MRT.

1.3 Related work

Fine-tuning general models on domain-specific datasets has become common in NMT. Simple transfer learning on new data can adapt a general model to in-domain data (Luong and Manning, 2015). Mixed fine-tuning where some original data is combined with the new data avoids reduced performance on the original data-set (Chu et al., 2017). We are only interested in performance on one domain, so use simple transfer learning.

For this task, we specifically fine-tune on a relatively small dataset. Adaptation to very small, carefully-chosen domains has been explored for speaker-personalized translation (Michel and Neubig, 2018) , and to reduce gender bias effects (Saunders and Byrne, 2020) while maintaining general domain performance. We wish to adapt to a very specific domain without need to maintain good general domain performance, but must avoid overfitting. Related approaches include fine-tuning a separate model for each test sentence (Li et al., 2018; Farajian et al., 2017) or test document (Xu et al., 2019; Kothur et al., 2018). We choose to train a single model for all test sentences in a language pair, but improve the robustness of that model to overfitting and exposure bias using MRT.

	Phase	Datasets	Sentence pairs	Dev datasets	Sentence pairs
en-es	Pre-training	UFAL Medical[1]	639K	Khresmoi[2]	1.5K
		Scielo[3]	713K		
		Medline titles[4]	288K		
		Medline abstracts	83K		
		Total	1723K / **1291K**		
	Fine-tuning	Medline abstracts	83K / **67.5K**	Biomedical19	800
en-de	Pre-training	UFAL Medical	2958K	Khresmoi	1.5K
		Medline abstracts	33K	Cochrane[5]	467
		Total	2991K / **2156K**		
	Fine-tuning	Medline abstracts	33K / **28.6K**	Biomedical19	800

Table 2: Biomedical training and validation data used in the evaluation task. For both language pairs identical data was used in both directions. Bolded numbers are totals after filtering

MRT has been widely applied to NMT in recent years (Shen et al., 2016; Neubig, 2016; Edunov et al., 2018). In particular, Wang and Sennrich (2020) recently highlighted the efficacy of MRT for reducing the effects of exposure bias.

2 Experimental setup

2.1 Data

We report on two language pairs: English-Spanish (en-es) and English-German (en-de). Table 2 lists the data used to train our biomedical domain evaluation systems. For each language pair we use the same training data in both directions, and preprocess all data with Moses tokenization, punctuation normalization and truecasing. We use a 32K-merge joint source-target BPE vocabulary (Sennrich et al., 2016) learned on the pre-training data.

All of our submitted approaches involve fine-tuning pre-trained models. We initialise fine-tuning with the strong biomedical domain models that formed our 'run 1' submission for the WMT19 biomedical translation task. Details of data preparation and training for these models are discussed in Saunders et al. (2019).

We fine-tune these models on Medline abstracts data, validating on test sets from the 2019 Biomedical task. For these we concatenate the src-trg and trg-src 2019 test sets for each language pair, and select only the 'OK' aligned sentences as annotated by the organizers.

Before fine-tuning we carry out detected language filtering on the Medline abstracts fine-tuning data using the Python LangDetect package[6]. We find LangDetect has a tendency to incorrectly label short sentences or those with rare vocabulary (very common in Medline) as a random language. For each language pair we therefore filter out only sentences where LangDetect identifies the source sentence as belonging to the target language, and vice versa.

We then use a series of simple heuristics to further filter the parallel datasets, removing duplicate sentence pairs, those with source/target length ratio of $<$ 1:3.5 or $>$ 3.5:1, and sentences with $>$ 120 tokens. For the more aggressively-filtered 'no-title' experiments we additionally remove all lines containing multiple tokens in square brackets, which in medical writing are used to denote the English translation of a non-English article's title (Patrias and Wendling, 2007). This leaves 27.3K sentence pairs for en-de and 64.8K for en-es: about 96% of the filtered data in both cases.

2.2 Model hyperparameters and training

We use the Tensor2Tensor implementation of the Transformer model with the `transformer_big` setup for all NMT models (Vaswani et al., 2018). We use the same effective batch size of 4k tokens for both MLE and doc-MRT. Because of model size constraints and the need to sample multiple targets for doc-MRT, we achieve the 4k effective batch size by accumulating gradients (Saunders et al., 2018) over every 4 batches of 1k tokens for MLE and every 16 batches of 256 tokens for doc-MRT.

For doc-MRT we use sampling temperature $\tau = 0.3$, smoothing parameter $\alpha = 0.6$ and $N = 8$ samples per sentence, which gave the best results for our doc-MRT experiments in Saunders et al. (2020).

[1] https://ufal.mff.cuni.cz/ufal_medical_corpus

[2] Dušek et al. (2017)

[3] Neves et al. (2016)

[4] https://github.com/biomedical-translation-corpora/medline (Yepes et al., 2017)

[5] http://www.himl.eu/test-sets

[6] https://pypi.org/project/langdetect/

		de2en	en2de	es2en	en2es
1	Baseline	38.8	30.6	48.5	46.6
2	MLE fine-tuning from 1	40.9	32.5	48.5	46.0
3	Checkpoint averaging 2 (en-de) / 1 (en-es)	41.1	32.2	48.5	47.1
4	MRT from 1	40.0	31.1	**49.0**	47.4
5	MRT from 2 (en-de only)	**41.3**	32.9	-	-
6	Checkpoint averaging 5 (en-de) / 4 (en-es)	**41.3**	**33.0**	48.9	**47.7**

Table 3: Validation BLEU developing models used in English-German and English-Spanish language pair submissions. Scores for single checkpoints unless indicated. MLE fine-tuning did not improve over the en-es baselines, so we do not use these models to initialise MRT.

	de2en	en2de	es2en	en2es
MLE from baseline	41.1	32.2	-	-
MLE from baseline, no-title	41.4	31.8	-	-
MRT from: MLE (en-de) / baseline (en-es)	41.3	**33.0**	48.9	**47.7**
MRT no-title from: MLE no-title (en-de) / baseline (en-es)	**41.9**	32.6	**49.0**	47.2

Table 4: Validation BLEU developing models used in English-German and English-Spanish language pair submissions. Scores for averaged checkpoints. MLE fine-tuning with either dataset did not improve over the en-es baselines.

For each approach we fine-tune on a single GPU, saving checkpoints every 1K updates, until fine-tuning validation set BLEU fails to improve for 3 consecutive checkpoints. Generally this took about 5K updates. We then perform checkpoint averaging (Junczys-Dowmunt et al., 2016) over the final 3 checkpoints to obtain the final model.

2.3 Inference

For the 2020 submissions, we additionally split any test lines containing multiple sentences before inference using the Python NLTK package[7], translate the split sentences separately, then remerged. We found this gave noticeable improvements in quality for the few sentences it applied to. In all cases we decode with beam size 4 using SGNMT (Stahlberg et al., 2017). Test scores are as provided by the organizers for "OK" sentences using Moses tokenization and the multi-eval tool. Validation scores are for case-insensitive, detokenized text obtained using SacreBLEU[8] (Post, 2018).

2.4 Results

We first assess the impact of small-domain adaptation to the full title-included Medline training set. Results in Table 3 show that small-domain MLE can lead to over-fitting and reduced performance (en-es) but also significant gains (en-de). Further fine-tuning with doc-MRT improved performance relative to the best MLE model for all translation directions by up to 0.8 BLEU when comparing with or without checkpoint averaging. While checkpoint averaging slightly decreased validation set performance for en2de MLE, we use it in all cases since it reduces sensitivity to randomness in training (Popel and Bojar, 2018).

In Table 4 we explore the impact of fine-tuning only on aggressively filtered 'no-title' data. This does noticeably improve performance for de2en, with a very small improvement for es2en. Since the added information in 'title' sentences is on the English side, this suggests that target training sentence quality impacts both MLE and MRT performance. However, removing these sentences entirely results in a noticeable performance decrease for the en2de and en2es models, demonstrating that they can be valuable training examples.

We submitted three runs to the WMT20 biomedical task for each language pair. For en-de run 1 was the baseline model fine-tuned on MLE with all data, while for en-es we submitted the checkpoint averaged baseline as MLE fine-tuning did not improve dev set performance. Run 2 was the run 1 model fine-tuned with doc-MRT on no-title data. Run 3 was the run 1 model fine-tuned with doc-MRT on all Medline abstract data. Table 5 gives scores for these submitted models.

Our best runs achieve the best and second-best results among all systems for en2es and es2en respectively as reported by the organizers. For en-de our test scores are further behind other systems, perhaps indicating that the baseline system could have been stronger before fine-grained adaptation.

[7] `https://pypi.org/project/nltk/` sentence splitter

[8] SacreBLEU signature: `BLEU+case.lc+numrefs.1 +smooth.exp+tok.13a+version.1.2.11`

	de2en		en2de		es2en		en2es	
	Dev	**Test**	**Dev**	**Test**	**Dev**	**Test**	**Dev**	**Test**
MLE (all data) (en-de) / Baseline (en-es)	41.1	39.6	32.2	32.9	48.5	**46.6**	47.1	45.7
MRT (no-title data)	**41.9**	39.6	32.6	32.8	**49.0**	46.4	47.2	**46.7**
MRT (all data)	41.3	**39.8**	**33.0**	**33.2**	48.9	**46.6**	47.7	46.6

Table 5: Validation and test BLEU for models used in English-German and English-Spanish language pair submissions. Test results are for "OK sentences" as scored by the organizers.

This is also indicated by the strong improvement of these models under simple MLE.

We submitted the MRT model on no-title data instead of the MLE on no-title data because MLE optimization did not improve over the baseline for en-es or en-es, with or without title lines, whereas MRT fine-tuning did. We also wanted to further examine whether MRT was robust enough to benefit from 'noisy' data like the title lines, or whether cleaner no-title training data was more useful. In fact both forms of doc-MRT performed similarly on the test data, except in the case of en2de, where 'no-title' MRT scored 0.4 BLEU worse – further confirmation that source sentences with more information than the gold target can benefit MRT. We note that a MRT run was the best run or tied best run in all cases.

For the test runs, we additionally experimented with simply removing square bracket tokens from source sentences, since these could act as 'triggering' tokens for title sentences. This did seem to improve translations for the sentences it applied to, but is clearly not applicable to all forms of exposure bias, since it requires knowledge of all behaviours that could trigger exposure bias. MRT does not require such knowledge, but still reduces the effects of exposure bias.

3 Conclusions

Our WMT20 Biomedical submission investigates improvements on the English-German and English-Spanish language pairs under a single strong model. In particular, we focus on the behaviour of models trained on sentences with some predictable irregularities. We find that aggressively filtering target sentences can help overall performance, but that aggressively filtering source sentence tends to hurt performance. We also find that Minimum Risk Training can benefit from imperfectly aligned training examples while reducing the effects of exposure bias.

Acknowledgments

This work was supported by EPSRC grants EP/M508007/1 and EP/N509620/1 and has been performed using resources provided by the Cambridge Tier-2 system operated by the University of Cambridge Research Computing Service[9] funded by EPSRC Tier-2 capital grant EP/P020259/1.

References

Rachel Bawden, Kevin Bretonnel Cohen, Cristian Grozea, Antonio Jimeno Yepes, Madeleine Kittner, Martin Krallinger, Nancy Mah, Aurelie Neveol, Mariana Neves, Felipe Soares, Amy Siu, Karin Verspoor, and Maika Vicente Navarro. 2019. Findings of the WMT 2019 biomedical translation shared task: Evaluation for MEDLINE abstracts and biomedical terminologies. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 29–53, Florence, Italy. Association for Computational Linguistics.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems*, pages 1171–1179.

Chenhui Chu, Raj Dabre, and Sadao Kurohashi. 2017. An empirical comparison of domain adaptation methods for neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 385–391.

Ondřej Dušek, Jan Hajič, Jaroslava Hlaváčová, Jindřich Libovický, Pavel Pecina, Aleš Tamchyna, and Zdeňka Urešová. 2017. Khresmoi summary translation test data 2.0. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Sergey Edunov, Myle Ott, Michael Auli, David Grangier, et al. 2018. Classical structured prediction losses for sequence to sequence learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, volume 1, pages 355–364.

[9] http://www.hpc.cam.ac.uk

M. Amin Farajian, Marco Turchi, Matteo Negri, and Marcello Federico. 2017. Multi-domain neural machine translation through unsupervised adaptation. In *Proceedings of the Second Conference on Machine Translation*, pages 127–137, Copenhagen, Denmark. Association for Computational Linguistics.

Marcin Junczys-Dowmunt, Tomasz Dwojak, and Rico Sennrich. 2016. The AMU-UEDIN submission to the WMT16 news translation task: Attention-based NMT models as feature functions in phrase-based SMT. In *Proceedings of the First Conference on Machine Translation*, pages 319–325, Berlin, Germany. Association for Computational Linguistics.

Sachith Sri Ram Kothur, Rebecca Knowles, and Philipp Koehn. 2018. Document-level adaptation for neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 64–73, Melbourne, Australia. Association for Computational Linguistics.

Xiaoqing Li, Jiajun Zhang, and Chengqing Zong. 2018. One sentence one model for neural machine translation. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Minh-Thang Luong and Christopher D Manning. 2015. Stanford Neural Machine Translation systems for spoken language domains. In *Proceedings of the International Workshop on Spoken Language Translation*, pages 76–79.

Paul Michel and Graham Neubig. 2018. Extreme adaptation for personalized neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 312–318, Melbourne, Australia. Association for Computational Linguistics.

Graham Neubig. 2016. Lexicons and minimum risk training for neural machine translation: NAIST-CMU at WAT2016. In *Proceedings of the 3rd Workshop on Asian Translation (WAT2016)*, pages 119–125, Osaka, Japan. The COLING 2016 Organizing Committee.

Mariana L Neves, Antonio Jimeno-Yepes, and Aurélie Névéol. 2016. The ScieLO Corpus: a Parallel Corpus of Scientific Publications for Biomedicine. In *LREC*.

Karen Patrias and Dan Wendling. 2007. Citing medicine: the nlm style guide for authors, editors. *and publishers. Bethesda, MD: National Library of Medicine. Retrieved June*, 27:2011.

Martin Popel and Ondřej Bojar. 2018. Training tips for the transformer model. *The Prague Bulletin of Mathematical Linguistics*, 110(1):43–70.

Matt Post. 2018. A call for clarity in reporting BLEU scores. *CoRR*, abs/1804.08771.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2016. Sequence level training with recurrent neural networks. In *ICLR*.

Danielle Saunders and Bill Byrne. 2020. Reducing gender bias in neural machine translation as a domain adaptation problem. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7724–7736, Online. Association for Computational Linguistics.

Danielle Saunders, Felix Stahlberg, and Bill Byrne. 2019. UCAM biomedical translation at WMT19: Transfer learning multi-domain ensembles. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 169–174, Florence, Italy. Association for Computational Linguistics.

Danielle Saunders, Felix Stahlberg, and Bill Byrne. 2020. Using context in neural machine translation training objectives. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7764–7770, Online. Association for Computational Linguistics.

Danielle Saunders, Felix Stahlberg, Adrià de Gispert, and Bill Byrne. 2018. Multi-representation ensembles and delayed sgd updates improve syntax-based nmt. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 319–325.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 1715–1725.

Shiqi Shen, Yong Cheng, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. 2016. Minimum Risk Training for Neural Machine Translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 1683–1692.

Felix Stahlberg, Eva Hasler, Danielle Saunders, and Bill Byrne. 2017. SGNMT–A Flexible NMT Decoding Platform for Quick Prototyping of New Models and Search Strategies. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 25–30.

Ashish Vaswani, Samy Bengio, Eugene Brevdo, Francois Chollet, Aidan N. Gomez, Stephan Gouws, Llion Jones, Łukasz Kaiser, Nal Kalchbrenner, Niki Parmar, Ryan Sepassi, Noam Shazeer, and Jakob Uszkoreit. 2018. Tensor2Tensor for Neural Machine Translation. *CoRR*, abs/1803.07416.

Chaojun Wang and Rico Sennrich. 2020. On exposure bias, hallucination and domain shift in neural machine translation. In *Proceedings of the 58th Annual*

Meeting of the Association for Computational Linguistics, pages 3544–3552, Online. Association for Computational Linguistics.

Jitao Xu, Josep Crego, and Jean Senellart. 2019. Lexical micro-adaptation for neural machine translation. In *International Workshop on Spoken Language Translation*.

Antonio Jimeno Yepes, Aurélie Névéol, Mariana Neves, Karin Verspoor, Ondrej Bojar, Arthur Boyer, Cristian Grozea, Barry Haddow, Madeleine Kittner, Yvonne Lichtblau, et al. 2017. Findings of the wmt 2017 biomedical translation shared task. In *Proceedings of the Second Conference on Machine Translation*, pages 234–247.

UoS Participation in the WMT20 Translation of Biomedical Abstracts

Felipe Soares
University of Sheffield - NLP Group
fs@felipesoares.net

Delton de Andrade Vaz
UFRGS
University of Montpellier
delton.vaz@gmail.comn

Abstract

This paper describes the machine translation systems developed by the University of Sheffield (UoS) team for the biomedical translation shared task of WMT20. Our system is based on a Transformer model with TensorFlow Model Garden toolkit. We participated in ten translation directions for the English/Spanish, English/Portuguese, English/Russian, English/Italian, and English/French language pairs. To create our training data, we concatenated several parallel corpora, both from in-domain and out-of-domain sources.

1 Introduction

In this paper, we present the system developed by the University of Sheffield for the Biomedical Translation shared task in the Fifth Conference on Machine Translation (WMT20), which consists in translating scientific texts from the biological and health domain.

Our participation in this task considered the English/Portuguese, English/Spanish, English/Russian, English/Italian, and English/French language pairs with translations in both directions. For that matter, we developed a machine translation (MT) system based on neural machine translation (NMT), using Google's TensorFlow Model Garden. [1]

2 Related Works

Previous participation in biomedical translation tasks include the works of Costa-Jussà et al. (2016) which employed Moses Statistic Machine Translation (SMT) to perform automatic translation integrated with a neural character-based recurrent neural network for model re-ranking and bilingual word embeddings for out of vocabulary (OOV) resolution. Given the 1000-best list of SMT translations, the RNN performs a re-scoring and selects the translation with the highest score. The OOV resolution module infers the word in the target language based on the bilingual word embedding trained on large monolingual corpora. Their reported results show that both approaches can improve BLEU scores, with the best results given by the combination of OOV resolution and RNN re-ranking. Similarly, Ive et al. (2016) also used the n-best output from Moses as input to a re-ranking model, which is based on a neural network that can handle vocabularies of arbitrary size.

More recently, Tubay and Costa-Jussà (2018) employed multi-source language translation using romance languages to translate from Spanish, French, and Portuguese to English. They used data from SciELO and Medline abstracts to train a Transformer model with individual languages to English and also with all languages concatenated to English.

In the last two WMT biomedical translation challenges (WMT18 and WMT19) (Neves et al., 2018; Bawden et al., 2019), the submissions that achieved the best BLEU scores for the ES/EN and PT/EN, in both directions (Soares and Becker, 2018; Tubay and Costa-Jussà, 2018; Carrino et al., 2019; Saunders et al., 2019; Soares and Krallinger, 2019), used the Transformer architecture with enhancements such as handling of terminology during tokenization (Carrino et al., 2019), multi-domain inference (Saunders et al., 2019) and exploitation of additional linguistic resources (Soares and Becker, 2018; Soares and Krallinger, 2019).

3 Resources

In this section, we describe the language resources used to train both models.

[1] https://github.com/tensorflow/models

Proceedings of the 5th Conference on Machine Translation (WMT), pages 870–874
Online, November 19–20, 2020.

3.1 Corpora

We used both in-domain and general domain corpora to train our systems. For general domain data, we used the ParaPat patent corpus (Soares et al., 2020), which is available for several languages, included the ones we explored in our systems. As for in-domain data, we included several different corpora:

- The corpus of full-text scientific articles from SciELO (Soares et al., 2018a), which includes articles from several scientific domains in the desired language pairs, but predominantly from biomedical and health areas.
- A subset of the UFAL medical corpus[2], containing the Medical Web Crawl data for the English/Spanish language pair.
- The EMEA corpus (Tiedemann, 2012), consisting of documents from the European Medicines Agency.
- A corpus of theses and dissertations abstracts (BDTD) (Soares et al., 2018b) from CAPES, a Brazilian governmental agency responsible for overseeing post-graduate courses. This corpus contains data only for the English/Portuguese language pair.
- A corpus from Virtual Health Library[3] (BVS), containing also parallel sentences for the language pairs explored in our systems.
- A corpus from SciELO (Neves et al., 2016), containing also parallel sentences from abstracts in English/Portuguese, English/Spanish, and English/French.

A new crawl of MEDLINE using the Ebot provided by the National Library of Medicine.[4]

Table 1 depicts the original number of parallel segments according to each corpora source. In Section 4.1, we detail the pre-processing steps performed on the data to comply with the task evaluation.

[2] https://ufal.mff.cuni.cz/ufal_medical_corpus

[3] http://bvsalud.org/

[4] https://www.ncbi.nlm.nih.gov/Class/PowerTools/eutils/ebot/ebot.cgi

4 Experimental Settings

In this section, we detail the pre-processing steps employed as well as the architecture of the Transformer.

4.1 Pre-processing

As detailed in the description of the biomedical translation task, the evaluation is based on texts extracted from MEDLINE. Since two of our corpora, the one comprised of full-text articles from SciELO and the new crawl from PubMed, may contain a considerable overlap with MEDLINE data, we decided to employ a filtering step in order to avoid including such data.

The first step in our filter was to download the parallel data from PubMed articles in Russian, French, and Italian. For that matter, we used the Ebot utility[5] provided by NLM using the queries *ITA[la]*, *FRE[la]*, and *RUS[la]*, retrieving all results available. Once downloaded, we performed sentence alignment using LF-Aligner[6]. To perform the filtering, we decided to use simple case insensitive string matching with *grep* supplying the option *-xvf* and the test set in English.

4.2 NMT System

As for the NMT system, we employed the official Google's implementation of the Transformer architecture (Vaswani et al., 2017) to train ten MT systems for the five language pairs. Tokenization was performed using the WordPiece unsupervised tokenizer with a vocabulary size of 32,000 on the initial training data, with a shared vocabulary between source and target.

For systems where the target language was English, back-translation was used with a number of sentences equals to the initial training system where English was the source. For the Spanish/English language pair, the system used to produce the artificial parallel sentences was the one developed by Soares and Krallinger (2019), while for the other language pairs we used the same systems trained by our team.

The parameters of our network for all language pairs excluding English/Portuguese are as follows. Encoder and Decoder: Transformer; Word vector size: 512; Layers for encoder and decoder:

[5] https://www.ncbi.nlm.nih.gov/Class/PowerTools/eutils/ebot/ebot.cgi

[6] https://sourceforge.net/projects/aligner/

Corpus	Sentences				
	EN/ES	EN/PT	EN/FR	EN/RU	EN/IT
ParaPat	-	-	-	3.28M	-
UFAL	286,779	-	1.6M	-	-
Abstract SciELO	767,069	669,629	-	-	-
Full-text SciELO	425,631	2.86M	-	-	-
EMEA	1.01M	1.08M	609,852	-	1.08M
CAPES-BDTD	-	950,252	-	-	-
BVS	-	931,946	10,812	-	-
MEDLINE (titles and abstracts)	-	-	582,007	11,271	1,298
Total	2.48M	6.49M	2.25M	3.28M	1.08M

Table 1: Original size of individual corpora used in our experiments

6; Attention heads: 16; RNN size: 512; Hidden transformer feed-forward: 2048; Batch size: 8196. For the English/Portuguese language pair, due to the large training set, we employed a bigger network as follows. Word vector size: 1024; Layers for encoder and decoder: 6; Attention heads: 16; RNN size: 1024; Hidden transformer feed-forward: 4096; Batch size: 8192.

To train our systems, we used 5 Tensor Processing Units (TPUs) v3, with a number of 250,000 steps (for all systems with exception of Russian, which was trained with fewer steps). The models with the best perplexity value were chosen as final models.

For the English/Russian language pair, incremental training was performed, since the size of the in-domain dataset was reduced. For such, we first trained our system in the out-of-domain data from patents for 100,000 steps. We then proceeded with additional training for 25,000 steps with in-domain data.

5 Results

We now detail the results achieved by our Transformer systems on the official test data used in the shared task regarding automatic evaluation. Table 2 shows the BLEU scores (Papineni et al., 2002) for our systems for the 10 language pairs we participated. For the Spanish and Portuguese language pairs we achieve high competitive results. For ES/EN, the best system (NLE) achieved BLEU of 0.5075, while the second best achieved BLEU of 0.4662 (TRAMECAT), very close to our result of 0.4624. For the opposite direction, EN/ES, the best system (UCAM) achieved 0.4662, the second best (Elhuyar_NLP) 0.4498, while our system scored 0.4493.

Language Pair	**BLEU**
EN/PT	0.4744
PT/EN	0.5334
EN/ES	0.4493
ES/EN	0.4624
EN/FR	0.3049
FR/EN	0.3514
EN/RU	0.2573
RU/EN	0.2936
EN/IT	0.2073
IT/EN	0.2276

Table 2: Official BLEU scores for the language pairs we submitted systems. These scores are evaluated on the "OK" aligned sentences.

For the Portuguese language, in both directions we achieved the best scores, with an EN/PT BLEU of 0.4744 and PT/EN of 0.5334. The second team in both languages (UNICAMP_DL) achieved scores of 0.4095 and 0.4988, respectively.

As for the Russian, French, and Italian languages, our scores were not as competitive as the best systems, with the exception of FR/EN, which we stood as 3 out of 5 teams. After carefully checking our training data, we found encoding issues with the different gathered data for those languages, especially with the encoding and tokenization of words containing apostrophes in French and Italian, as well as the Cyrillic Kha.

6 Conclusions

We presented the University of Sheffield (UoS) machine translation system for the biomedical translation shared task in WMT20. For our submission, we trained ten Transformers NMT systems, employing different corpora for each language pair. In addition, for systems with English as target language, back-translation was used, and for the Russian language, incremental training from Patent abstracts was used.

For model building, we included several corpora from biomedical and health domain, and from out-of-domain data that we considered to have similar textual structure, such as books and patents. Prior training, we also pre-processed our corpora to ensure that we did not include any sentence from the released test set, which could produce biased models.

Regarding future work, we are planning on optimizing our systems by performing pre-selection of out-of-domain data, aiming at selecting only the most similar sentences to the in-domain data. In addition, we plan to explore the potential use of domain-specific decoding, as proposed in Saunders et al. (2019).

Acknowledgements

This work was supported by Amazon AWS Cloud Credits for Research, which were used for corpora processing and gathering, and by Google TensorFlow Research Cloud credits, which were used for model training and inference.

References

Rachel Bawden, Kevin Bretonnel Cohen, Cristian Grozea, Antonio Jimeno Yepes, Madeleine Kittner, Martin Krallinger, Nancy Mah, Aurelie Neveol, Mariana Neves, Felipe Soares, Amy Siu, Karin Verspoor, and Maika Vicente Navarro. 2019. Findings of the wmt 2019 biomedical translation shared task: Evaluation for medline abstracts and biomedical terminologies. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 31–55, Florence, Italy. Association for Computational Linguistics.

Casimiro Pio Carrino, Bardia Rafieian, Marta R. Costa-jussà, and JosÃ© A. R. Fonollosa. 2019. Terminology-aware segmentation and domain feature for the wmt19 biomedical translation task. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 153–157, Florence, Italy. Association for Computational Linguistics.

Marta R Costa-Jussà, Cristina España-Bonet, Pranava Madhyastha, Carlos Escolano, and José AR Fonollosa. 2016. The talp–upc spanish–english wmt biomedical task: Bilingual embeddings and char-based neural language model rescoring in a phrase-based system. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, volume 2, pages 463–468.

Julia Ive, Aurélien Max, and François Yvon. 2016. Limsi's contribution to the wmt'16 biomedical translation task. In *First Conference on Machine Translation*, volume 2, pages 469–476.

Mariana Neves, Antonio Jimeno Yepes, Aurélie Névéol, Cristian Grozea, Amy Siu, Madeleine Kittner, and Karin Verspoor. 2018. Findings of the wmt 2018 biomedical translation shared task: Evaluation on medline test sets. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 328–343, Belgium, Brussels. Association for Computational Linguistics.

Mariana Neves, Antonio Jimeno Yepes, and Aurélie Névéol. 2016. The scielo corpus: a parallel corpus of scientific publications for biomedicine. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France. European Language Resources Association (ELRA).

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Danielle Saunders, Felix Stahlberg, and Bill Byrne. 2019. Ucam biomedical translation at wmt19: Transfer learning multi-domain ensembles. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 171–176, Florence, Italy. Association for Computational Linguistics.

Felipe Soares and Karin Becker. 2018. Ufrgs participation on the wmt biomedical translation shared task. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 673–677, Belgium, Brussels. Association for Computational Linguistics.

Felipe Soares and Martin Krallinger. 2019. Bsc participation in the wmt translation of biomedical abstracts. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 177–180, Florence, Italy. Association for Computational Linguistics.

Felipe Soares, Viviane Moreira, and Karin Becker. 2018a. A Large Parallel Corpus of Full-Text Scientific Articles. In *Proceedings of the Eleventh International Conference on Language Resources and*

Evaluation (LREC 2018), Miyazaki, Japan. European Language Resources Association (ELRA).

Felipe Soares, Mark Stevenson, Diego Bartolome, and Anna Zaretskaya. 2020. ParaPat: The multi-million sentences parallel corpus of patents abstracts. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 3769–3774, Marseille, France. European Language Resources Association.

Felipe Soares, Gabrielli Yamashita, and Michel Anzanello. 2018b. A parallel corpus of theses and dissertations abstracts. In *The 13th International Conference on the Computational Processing of Portuguese (PROPOR 2018)*, Canela, Brazil. Springer International Publishing.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey. European Language Resources Association (ELRA).

Brian Tubay and Marta R. Costa-Jussà. 2018. Neural machine translation with the transformer and multi-source romance languages for the biomedical wmt 2018 task. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 678–681, Belgium, Brussels. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Ixamed's submission description for WMT20 Biomedical shared task: benefits and limitations of using terminologies for domain adaptation

Xabier Soto, Olatz Perez-de-Viñaspre, Gorka Labaka, Maite Oronoz
HiTZ Basque Center for Language Technologies - Ixa, University of the Basque Country UPV/EHU
{xabier.soto, olatz.perezdevinaspre, gorka.labaka, maite.oronoz }@ehu.eus

Abstract

In this paper we describe the systems developed at Ixa for our participation in WMT20 Biomedical shared task in three language pairs, en-eu, en-es and es-en. When defining our approach, we have put the focus on making an efficient use of corpora recently compiled for training Machine Translation (MT) systems to translate Covid-19 related text, as well as reusing previously compiled corpora and developed systems for biomedical or clinical domain. Regarding the techniques used, we base on the findings from our previous works for translating clinical texts into Basque, making use of clinical terminology for adapting the MT systems to the clinical domain. However, after manually inspecting some of the outputs generated by our systems, for most of the submissions we end up using the system trained only with the basic corpus, since the systems including the clinical terminologies generated outputs shorter in length than the corresponding references. Thus, we present simple baselines for translating abstracts between English and Spanish (en/es); while for translating abstracts and terms from English into Basque (en-eu), we concatenate the best en-es system for each kind of text with our es-eu system. We present automatic evaluation results in terms of BLEU scores, and analyse the effect of including clinical terminology on the average sentence length of the generated outputs. Following the recent recommendations for a responsible use of GPUs for NLP research, we include an estimation of the generated CO_2 emissions, based on the power consumed for training the MT systems.

1 Introduction

The WMT20 Biomedical shared task calls for developing systems for translating biomedical abstracts and terminologies between several languages. In our case, we participate in the task of translating biomedical terms and abstracts from English into Basque (en-eu), as well as translating biomedical abstracts between English and Spanish (en-es and es-en). For translating the test data from English into Basque, we concatenate our best en-es system with our es-eu system, both for translating abstracts and terminologies.

2 Related work

For translating biomedical texts from English into Catalan, Costa-jussá et al. (2018) use a pivoting or cascade approach, translating the texts first from English into Spanish (en-es), and then from Spanish into Catalan (es-ca). This technique is useful when there are more bilingual in-domain sentences for each of the language pairs (en/es and es/ca) than for the desired source and target languages (en/ca). Since there are low resources for en/eu biomedical domain, but we have access to many resources for en/es and es/eu in the biomedical or clinical domain, we follow the same approach for translating the test sets from English into Basque (en-eu).

Since most of the available in-domain corpus is monolingual, we also make use of traditional back-translation and forward translation techniques (Sennrich et al., 2016).

In our previous work for translating clinical texts between Basque and Spanish, we showed that including clinical terminologies directly into the training corpus was useful for domain adaptation when no bilingual in-domain sentences were available (Soto et al., 2019a). As clinical terminologies, we refer to the automatic translation into Basque of SNOMED CT (IHTSDO, 2014), which is considered the most comprehensive, multilingual clinical health care terminology collection in the world. In this work, we extend the number of clinical terminologies as part of the ongoing translation of SNOMED CT into Basque (Perez-de-Viñaspre,

Proceedings of the 5th Conference on Machine Translation (WMT), pages 875–880

2017), and include the provided ICD-10 resources plus other smaller terminology collections recently created for translating Covid-19 related texts.

3 Resources

For training our baseline en/es systems, we make use of the Medline corpus provided by the organisers of the WMT20 Biomedical shared task, as well as the recently compiled TAUS Corona Crisis Corpus.[1]

For backtranslation (es-en) and forward translation (en-es), we use the English corpus prepared by Sketch Engine[2], based on the Covid-19 related corpus compiled for a recent Kaggle competition (Wang et al., 2020).

As a final step, we include several clinical terminologies: 1) the ICD-10 (en-eu) corpus provided by the organisers of the WMT20 Biomedical shared task, adding the corresponding Spanish counterparts; 2) terms obtained from the automatic translation into Basque of SNOMED CT (Perez-de-Viñaspre, 2017), including terms up to 11 tokens; 3) a recent SNOMED CT interim release of Covid-19 related terms[3], manually translated into Basque by a translator of the Basque public health service (Osakidetza); and 4) a collection of Covid-19 related terms recently compiled by Elhuyar[4], including all the terms published until June 18[5].

For training our es-eu system, we use the aforementioned terminologies together with an out-of-domain corpus formed mainly by news (Etchegoyhen et al., 2016), previously applying a language identification tool[6] to exclude sentences where most of the terms are named entities like locations or person names. Doing this, a bigger part of the vocabulary can be used to translate biomedical or clinical terms. Furthermore, as in-domain corpus we use clinical notes in Spanish coming from the hospital of Galdakao-Usansolo for forward translation and copying (Currey et al., 2017). This corpus was compiled between 2008 and 2012.[7]

For the evaluation of en/es systems, we use Khresmoi;[8] while for es-eu we use templates of clinical notes in Basque written in the Donostia hospital (Joanes Etxeberri Saria V. Edizioa, 2014), together with their manual translations into Spanish made by a bilingual doctor.

Table 1 presents the description and statistics of our corpora.

	Description	Sentences
en/es	Medline (WMT Biomedical)	388,068
	TAUS Corona Crisis Corpus	902,133
	Sketch Engine Covid-19 (en)	4,671,609
	ICD-10 (WMT Biomedical)	27,696
	SNOMED CT corpus	385,800
	SNOMED CT Covid-19 corpus	84
	Elhuyar Covid-19 corpus	113
	Khresmoi (dev set)	500
	Khresmoi (test set)	1,000
es-eu	out-of-domain	3,703,757
	in-domain (es)	2,023,811
	ICD-10 (WMT Biomedical)	27,696
	SNOMED CT corpus	896,898
	SNOMED CT Covid-19 corpus	84
	Elhuyar Covid-19 corpus	126
	Donostia hospital (dev set)	1,038
	Donostia hospital (test set)	1,038

Table 1: Description and statistics of the used corpora.

4 Systems

For en/es we develop 3 systems: 1) using only the bilingual in-domain corpus (Medline + TAUS Corona Crisis Corpus), 2) including the Sketch Engine Covid-19 (en) corpus for backtranslation (es-en) or forward translation (en-es), and 3) adding all the clinical terminologies from ICD-10, SNOMED CT and Elhuyar.

For es-eu we train a unique system using the out-of-domain corpus and the clinical terminologies, as well as the in-domain (es) corpus for forward translation and copying.

For training the backtranslation (en-es) and forward translation (es-en) systems, we used the bilingual in-domain corpus (Medline + TAUS Corona Crisis Corpus); while for es-eu we used the out-of-domain corpus and a reduced set of SNOMED CT terminologies, as used in Soto et al. (2019b).

[1] https://md.taus.net/corona

[2] https://www.sketchengine.eu/covid19/

[3] http://www.snomed.org/news-and-events/articles/march-2020-interim-snomedct-release%2DCOVID-19

[4] https://www.elhuyar.eus/eu/site/prentsa-aretoa/368/covid-19-gaitzaren-inguruko-terminologia%2Dgure-hiztegietako-azkenaldaketak

[5] when the English term was missing, if there was no doubt about how to translate it, the first author manually translated it; while if there wasn't a clear translation into English or the term was more related to socioeconomics than biomedical domain, it wasn't included in the en/es corpus.

[6] https://github.com/saffsd/langid.py

[7] Due to privacy requirements, this corpus is not publicly available. Prior to use, it was de-identified by reordering sentences, and only authors who had previously signed a non-disclosure commitment had access to it.

[8] https://lindat.mff.cuni.cz/repository/xmlui/handle/11234/1-2122

All the systems are Transformer (Vaswani et al., 2017) models trained with OpenNMT (Klein et al., 2017), using the recommended hyperparameters.[9] When necessary, we halved the batch-size so that it could fit in 2 GPUs, and accordingly doubled the value for gradient accumulation.

We applied joint BPE-dropout (Provilkov et al., 2020), with 32,000 merge operations for en/es and 90,000 for es-eu.

5 Results

Table 2 shows the BLEU scores of our systems on the validation (dev) and test sets presented in Table 1, together with previously published (es-eu) results for comparison.

Lang.	System	dev	test
es-en	Baseline (Medline + TAUS)	56.57	52.55
	Baseline + backtranslation (bt)	**61.60**	**57.25**
	Baseline + bt + terminologies	60.95	56.89
en-es	Baseline (Medline + TAUS)	48.02	46.30
	Baseline + forward translation (ft)	**50.20**	**47.19**
	Baseline + ft + terminologies	49.92	47.15
es-eu	Soto et al. (2019a)	11.30	**12.04**
	Soto et al. (2019b)	**11.85**	11.24
	This work	6.21	5.15

Table 2: BLEU scores for systems developed for es-en, en-es and es-eu translation directions (Lang.).

As expected, backtranslation significantly improves the es-en results (around 5 BLEU points); while the gains obtained with forward translation (en-es) are smaller (around 2 BLEU points in the dev set and around 1 BLEU point in the test set). However, we observe a slight decrease on BLEU values when including the clinical terminologies on the training corpus for both es-en and en-es systems. For further analysing this, we calculate the average sentence length of the different evaluation corpora as translated by the different systems. Table 3 shows the average sentence length of the validation (dev) and test sets after being translated by each of the es-en and en-es systems. As a reference, the average sentence length of the original dev and test sets are 22.70 (es) / 21.06 (en) and 24.03 (es) / 21.91 (en).

We observe that, except for the dev set translated by the en-es systems, the lower sentence length is always obtained when using the system including the clinical terminologies. This is confirmed by a fast check of the outputs generated when translating

Lang.	System	dev	test
es-en	Baseline (Medline + TAUS)	20.54	22.02
	Baseline + backtranslation (bt)	20.56	21.73
	Baseline + bt + terminologies	20.40	21.56
en-es	Baseline (Medline + TAUS)	22.75	23.87
	Baseline + forward translation (ft)	22.93	23.84
	Baseline + ft + terminologies	22.99	23.76

Table 3: Average sentence length of the different evaluation corpora as translated by the systems developed for es-en and en-es translation directions (Lang.).

the official test sets provided by the organisers, where we see that the sentences translated by these systems usually end before having translated all of the terms that appear in the input. Overall, the sentence lengths of the generated translations are closer to the original sentence lengths when using the baseline systems; therefore, for en-es and es-en we submit as best systems the translations produced by the baseline systems, using only Medline and TAUS corpora.

Regarding es-eu, in Table 2 we can see a severe decrease on BLEU scores comparing to our previous works. For training the system in Soto et al. (2019a) we used the same out-of-domain corpus (without applying langid.py) and a reduced set of SNOMED CT terminologies (151,111 entries), both directly and inserted into artificial sentences; while in Soto et al. (2019b) we used this same corpus without the artificial sentences, which didn't prove to be useful. Nevertheless, after manually checking the outputs generated by these 3 systems, we observe that the system developed for this work performs generally better, so we submit the translations produced by this system.[10] As we use a cascade approach for en-eu, we use the en-es system including the terminologies for translating abstracts; and the baseline system for translating terminologies, as these were the best performing systems on a fast human evaluation. [11]

Once we have selected the best performing systems for each of the language pairs, since we are allowed to submit 3 runs, in the case of en/es, for each of the developed systems we submit an ensemble of the 3 models which obtained higher BLEU

[9] http://opennmt.net/OpenNMT-py/FAQ.html#how-do-i-use-the-transformer-model (Accessed on July 18, 2020.)

[10] It has to be noted that the evaluation corpus used for es-eu has strong limitations, since the original sentences are written for encouraging medicine students to write correctly; while the translations into Basque made by a doctor are overall shorter, use simplified grammar, often omit verbs and punctuation, and use many acronyms.

[11] Both for en/es and en-eu systems, the translations of the first 10 sentences of the official test sets were checked; and in case of tie, the next 10 sentences were also observed.

scores in the dev set during training; while for en-eu we alternate between single and ensemble systems for each of the en-es and es-eu systems. Specifically, we submit as best system an ensemble of the baseline en-es system and a single es-eu system for translating terminologies; while we use a single en-es system including the terminologies and an ensemble es-eu system for translating abstracts.

Table 4 shows the BLEU scores obtained on the official test sets for each of the language pairs and submitted runs for translating abstracts, as provided by the organisers. We present in italics the result of the expected best system for each language pair, and in bold the highest BLEU score, as in previous tables.

Lang.	System	BLEU
es-en	Baseline (Medline + TAUS)	*40.65*
	Baseline + backtranslation (bt)	**40.71**
	Baseline + bt + terminologies	39.96
en-es	Baseline (Medline + TAUS)	***41.71***
	Baseline + forward translation (ft)	38.36
	Baseline + ft + terminologies	38.58
en-eu	single (en-es) + ensemble (es-eu)	*8.15*
	ensemble (en-es) + single (es-eu)	7.82
	ensemble (en-es) + ensemble (es-eu)	**8.84**

Table 4: BLEU scores on the official test sets for translating abstracts in es-en, en-es and en-eu translation directions (Lang.).

Comparing to the submissions made by other teams, our systems submitted for en/es obtain the lowest BLEU scores among all the participants; while for en-eu our best run is the second among the best runs of each participant, only surpassed by the three runs submitted by Elhuyar.

Finally, Table 5 presents the accuracy and BLEU scores obtained by our systems when used for translating terminologies (en-eu), as provided by the organisers.

Lang.	System	Acc.	BLEU
en-eu	single (en-es) + ensemble (es-eu)	0.12	13.14
	ensemble (en-es) + single (es-eu)	*0.08*	*7.21*
	ensemble (en-es) + ensemble (es-eu)	**0.13**	**14.81**

Table 5: Accuracy (Acc.) and BLEU scores on the official test set for translating terminologies in en-eu translation direction (Lang.).

Surprisingly, the obtained automatic scores are much lower than the ones obtained by the rest of the participants (between 0.73 and 0.78 for accuracy, and approximately 71 to 74 BLEU scores). However, the generated translations look quite sensible, so we expect the human evaluation will shed some light about the performance of our systems.

6 Measured power consumption and estimated CO_2 emissions

Following the recommendations by Strubell et al. (2019), we report the power consumed by our GPUs when training the systems developed for this work, along with the estimated CO_2 emissions. For calculating the training time, we use the time shown in the first and last lines of the log file generated while training the systems, including also the initial time for preparing the data, so the presented values constitute an upper bound of the actually consumed power. Nonetheless, we have to point out that OpenNMT makes an efficient use of the power capabilities of the GPUs, so we can say that the numbers shown here are an accurate estimation. Table 6 shows the number of GPUs, training time, power consumption and estimated CO_2 emissions for each of the developed systems. All the GPUs used for this work are Nvidia Titan Xp models with 250W power. We present the values of the different systems in the same order as in Table 2, and estimate the CO_2 emissions by applying equations (1) and (2) in Strubell et al. (2019), considering only the power consumed by our GPUs. Overall, the CO_2 emissions generated by our GPUs are approximately 329.44 lbs.

Lang.	GPUs	Time (hh:mm)	Power (kWh)	CO_2e (lbs)
es-en	4	43:19	43.33	65.31
	2	46:30	23.26	35.06
	2	45:37	22.82	34.39
en-es	4	45:09	45.16	68.07
	2	47:24	23.70	35.73
	2	47:21	23.68	35.69
es-eu	2	73:14	36.62	55.20
TOTAL				**329.44**

Table 6: Number of GPUs, training time, power consumption and estimated CO_2 emissions for each of the developed systems (same order as in Table 2).

7 Conclusion and future work

In this work, we have presented a simple proposal using previously compiled corpora from the biomedical or clinical domain, as well as clinical terminology included directly to the training corpora. Apart from calculating BLEU scores, we have also calculated the average sentence length of the generated translations for en/es systems, and observed that the systems including terminologies

performed generally worse than the baseline systems.

As future work, we plan to incorporate these clinical terminologies in a more efficient way (Dinu et al., 2019; Wang et al., 2019). For improving both training and evaluation, we'll also use bilingual clinical domain corpora being compiled now in collaboration with the Basque public health service (Osakidetza). Furthermore, since we have observed that some of the translations generated by the es-eu systems remain in Spanish, we'll study techniques to leverage in-domain monolingual data in Basque like the one provided by the organisers from Wikipedia.

Finally, we plan to keep reporting the consumed power and consequently generated CO_2 emissions, probably making use of recently developed automatic tools (Henderson et al., 2020)[12].

Acknowledgments

This work was supported by the Spanish Ministry of Economy and Competitiveness (MINECO) FPI grant number BES-2017-081045, and projects BigKnowledge (BBVA foundation grant 2018), DOMINO (PGC2018-102041-B-I00, MCIU/AEI/FEDER, UE) and DOTT-HEALTH (PID2019-106942RB-C31, MCIU/AEI/FEDER, UE).

References

Marta R. Costa-jussá, Noé Casas, and Maite Melero. 2018. English-catalan neural machine translation in the biomedical domain through the cascade approach. *Computing Research Repository*, arXiv:1803.07139. Version 2.

Anna Currey, Antonio Valerio Miceli Barone, and Kenneth Heafield. 2017. Copied monolingual data improves low-resource neural machine translation. In *Proceedings of the Second Conference on Machine Translation*, pages 148–156.

Georgiana Dinu, Prashant Mathur, Marcello Federico, and Yaser Al-Onaizan. 2019. Training neural machine translation to apply terminology constraints. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3063–3068, Florence, Italy. Association for Computational Linguistics.

Thierry Etchegoyhen, Andoni Azpeitia, and Naiara Pérez. 2016. Exploiting a large strongly comparable corpus. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 3523–3529, Portoroz, Slovenia.

Peter Henderson, Jieru Hu, Joshua Romoff, Emma Brunskill, Dan Jurafsky, and Joelle Pineau. 2020. Towards the systematic reporting of the energy and carbon footprints of machine learning. *Computing Research Repository*, arXiv:2002.05651.

International Health Terminology Standards Development Organisation IHTSDO. 2014. *SNOMED CT Starter Guide*. Technical report, International Health Terminology Standards Development Organisation.

Joanes Etxeberri Saria V. Edizioa. 2014. Donostia unibertsitate ospitaleko alta-txostenak. *Donostiako Unibertsitate Ospitalea, Komunikazio Unitatea*.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics-System Demonstrations*, pages 67–72, Vancouver, Canada.

Olatz Perez-de-Viñaspre. 2017. *Automatic medical term generation for a low-resource language: translation of SNOMED CT into Basque*. Ph.D. thesis, University of the Basque Country, Donostia, Spain.

Ivan Provilkov, Dmitrii Emelianenko, and Elena Voita. 2020. BPE-dropout: Simple and effective subword regularization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1882–1892, Online. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Xabier Soto, Olatz Perez-De-Viñaspre, Gorka Labaka, and Maite Oronoz. 2019a. Neural machine translation of clinical texts between long distance languages. *Journal of the American Medical Informatics Association*, 26(12):1478–1487.

Xabier Soto, Olatz Perez-De-Viñaspre, Maite Oronoz, and Gorka Labaka. 2019b. Leveraging SNOMED CT terms and relations for machine translation of clinical texts from Basque to Spanish. In *Proceedings of the Second Workshop on Multilingualism at the Intersection of Knowledge Bases and Machine Translation*, pages 8–18, Dublin, Ireland.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and policy considerations for deep learning in nlp. *Computing Research Repository*, arXiv:1906.02243.

[12] https://github.com/Breakend/experiment-impact-tracker

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, Long Beach, CA.

Lucy Lu Wang, Kyle Lo, Yoganand Chandrasekhar, Russell Reas, Jiangjiang Yang, Doug Burdick, Darrin Eide, Kathryn Funk, Yannis Katsis, Rodney Kinney, Yunyao Li, Ziyang Liu, William Merrill, Paul Mooney, Dewey Murdick, Devvret Rishi, Jerry Sheehan, Zhihong Shen, Brandon Stilson, Alex Wade, Kuansan Wang, Nancy Xin Ru Wang, Chris Wilhelm, Boya Xie, Douglas Raymond, Daniel S. Weld, Oren Etzioni, and Sebastian Kohlmeier. 2020. Cord-19: The covid-19 open research dataset. *Computing Research Repository*, arXiv:2004.10706. Version 4.

Tao Wang, Shaohui Kuang, Deyi Xiong, and António Branco. 2019. Merging external bilingual pairs into neural machine translation. *Computing Research Repository*, arXiv:1912.00567.

Tencent AI Lab Machine Translation Systems for the WMT20 Biomedical Translation Task

Xing Wang, Zhaopeng Tu, Longyue Wang, Shuming Shi
Tencent AI Lab, Shenzhen, China
{brightxwang,zptu,vinnylywang,shumingshi}@tencent.com

Abstract

This paper describes the Tencent AI Lab submission of the WMT2020 shared task on biomedical translation in four language directions: German⇒English, English⇒German, Chinese⇒English and English⇒Chinese. We implement our system with model ensemble technique on different transformer architectures (DEEP, HYBRID, BIG, LARGE Transformers). To enlarge the in-domain bilingual corpus, we use back-translation of monolingual in-domain data in the English language as additional in-domain training data. Our systems in German⇒English and English⇒German are ranked 1st and 3rd respectively according to the official evaluation results in terms of BLEU scores.[1]

1 Introduction

Neural machine translation (Bahdanau et al., 2015; Vaswani et al., 2017, NMT) has achieved great progress in recent years. However, as Koehn and Knowles (2017) pointed out, NMT systems suffer from poor translation performance in out-of-domain scenarios, which poses a great challenge for the biomedical translation task.

In this paper, we present our submission to the WMT20 shared task on biomedical translation task. We participated in two language directions: German-English and Chinese-English. To address the domain problem, on one hand, we adopt model ensemble technique (Liu et al., 2018) with different transformer architectures to build a more robust model. On the other hand, we enlarge the in-domain bilingual corpus with back-translation approach (Sennrich et al., 2016a).

Our contributions are as follows:

- We adopt the model ensemble technique and the back-translation approach to achieve the state-of-the-art performance on WMT19 biomedical translation task test sets.
- To promote further studies, we release some pre-trained models and the in-domain synthetic Chinese-English bilingual data for the community.

The rest of this paper is organized as follows. Section 2 presents our system with four different transformer architectures: DEEP, HYBRID, BIG, LARGE Transformers. Section 3 describes the training data used in our system, including bilingual data, monolingual data and synthetic bilingual data. Section 4 reports experimental results in two language directions. Finally, we conclude our work in Section 5.

2 System

In our systems, we adopt four different model architectures with TRANSFORMER (Vaswani et al., 2017):

- **DEEP** TRANSFORMER (Dou et al., 2018; Wang et al., 2019a; Dou et al., 2019) is the TRANSFORMER-BASE model with the 40-layer encoder.
- **HYBRID** TRANSFORMER (Hao et al., 2019b) is the TRANSFORMER-BASE model with 40-layer hybrid encoder. The 40-layer hybrid encoder stacks 35-layer self-attention-based encoder on top of 5-layer bi-directional ON-LSTM (Shen et al., 2019) encoder.
- **BIG** TRANSFORMER is the TRANSFORMER-BIG model as used by Vaswani et al. (2017).
- **LARGE** TRANSFORMER is similar to TRANSFORMER-BIG model except that it uses a 20-layer encoder.

[1]Details of our systems are introduced in https://github.com/hsing-wang/WMT2020_BioMedical

Proceedings of the 5th Conference on Machine Translation (WMT), pages 881–886
Online, November 19–20, 2020.

	DEEP	HYBRID	BIG	LARGE
Encoder Layer	40	40	6	20
Decoder Layer	6	6	6	6
Attention Heads	8	8	16	16
Embedding Size	512	512	1024	1024
FFN Size	2048	2048	4096	4096

Table 1: Hyper-parameters of different Transformer models used in our system.

The main differences between these models are presented in Table 1. Pre-Norm (Wang et al., 2019a) is adopted in above four models. All models are implemented on top of the open-source toolkit Fairseq[2]. Model ensemble is used through ensemble decoding with different model architectures.

3 Data

The data used to train our system consists of three parts: bilingual data, monolingual data and synthetic bilingual data.

3.1 Bilingual Data

In-domain bilingual data The in-domain bilingual data is provided by WMT20 biomedical translation shared task. For German-English, we choose Biomedical Translation[3] and UFAL Medical Corpus[4] to use as the in-domain training data. For Chinese-English out-of-domain (OOD) data, we adopt data selection (Axelrod et al., 2011; Liu et al., 2014) to select the in-house data (8.5M sentence pairs) as the in-domain training data.

General-domain bilingual data To alleviate the data scarce problem, we collect general-domain bilingual data from WMT20 news translation shared task[5]. For German-English, we use Europarl-v10[6], ParaCrawl-v5.1[7], News Commentary-v15[8] and Wiki Titles-v2[9]. For Chinese-English, we use CCMT Corpus[10], UN Parallel Corpus v1.0[11], News Commentary-v15[12].

3.2 Monolingual Data

As WMT20 biomedical translation shared task provides in-domain bilingual data in other language pairs, we gather in-domain monolingual data from bilingual data in other language pair. Specifically, we collect the English side of the bilingual sentence pairs from Biomedical Translation and UFAL Medical Corpus.

The statistics of the in-domain bilingual and monolingual data is listed in Table 2.

3.3 Synthetic Bilingual Data

To enlarge the in-domain bilingual corpus, we adopt back-translation method (Sennrich et al., 2016a) to generate synthetic bilingual sentence pairs. For Chinese-English, as we lack of sufficient in-domain bilingual data, we use an online translation system TranSmart[13] to translate the in-domain monolingual English back to Chinese. For German-English, we train a English-German LARGE model on the combination of in-domain and general-domain bilingual data, and use the model to generate synthetic bilingual data.

4 Experiment

We report experimental results in four language pairs: German-English (de/en), English-German (en/de), Chinese-English (zh/en) and English-Chinese (en/zh).

4.1 Experimental Setup

Data Pre-Processing We follow previous work (Saunders et al., 2019; Peng et al., 2019) to

[2] https://github.com/pytorch/fairseq (Ott et al., 2019)
[3] https://github.com/biomedical-translation-corpora/corpora
[4] https://ufal.mff.cuni.cz/ufal_medical_corpus
[5] http://www.statmt.org/wmt18/translation-task.html
[6] http://www.statmt.org/europarl/v10/
[7] https://www.paracrawl.eu/index.php
[8] http://data.statmt.org/wikititles/v2/
[9] http://data.statmt.org/news-commentary/v15/
[10] http://mteval.cipsc.org.cn:81/agreement/description
[11] https://conferences.unite.un.org/UNCorpus/
[12] http://data.statmt.org/wikititles/v2/
[13] transmart.qq.com

Corpus	File	Zh/En	De/En	En
Biomedical Translation	wmt18training/es-en	n/a	n/a	287,811
	wmt18training/fr-en	n/a	n/a	627,576
	wmt18training/pt-en	n/a	n/a	74,645
	wmt19training/de-en	n/a	40,398	40,398
	wmt19training/fr-en	n/a	n/a	75,049
	wmt19training/es-en	n/a	n/a	100,257
	wmt19training/pt-en	n/a	n/a	49,918
	wmt20training/it-en	n/a	n/a	14,756
	wmt20training/ru-en	n/a	n/a	46,782
UFAL Medical Corpus	shuffled.de-en	n/a	37,814,533	37,814,533
	shuffled.cs-en	n/a	n/a	48,243,170
	shuffled.es-en	n/a	n/a	92,999,169
	shuffled.fr-en	n/a	n/a	88,526,658
	shuffled.hu-en	n/a	n/a	48,783,611
	shuffled.pl-en	n/a	n/a	39,442,076
	shuffled.ro-en	n/a	n/a	62,034,179
	shuffled.sv-en	n/a	n/a	23,142,661

Table 2: The detailed statistics of in-domain training data used in our system. "Zh/En" and "De/En" denote the Chinese-English and German-English bilingual data, respectively. "En" denotes the monolingual English data.

use Moses scripts[14] to preprocess[15] the data and filter the bilingual data with following heuristics rules:

- Filter out duplicate sentence pairs (Khayrallah and Koehn, 2018; Ott et al., 2018).
- Filter out sentence pairs with wrong language (Khayrallah and Koehn, 2018).
- Filter out sentences pairs containing more than 120 tokens or fewer than 3.
- Filter out sentence pairs with source/target length ratio exceeding 1.5 (Ott et al., 2018).

4.2 Evaluation

For German-English, we use the Khresmoi development data as the development set, and use the sentence pairs with the correct alignment in WMT19 biomedical translation task test set as our test set. For Chinese-English, we use the in-house bilingual test set (1,000 sentence pairs) and the sentence pairs with the correct alignment in WMT19 biomedical translation task test set as development set and test set, respectively.

Follow Bawden et al. (2019), we use multi-bleu.perl from Moses[16] to compute BLEU scores and report case-sensitive BLEU scores on development and test sets.

Data Pre-processing For each language pair, we perform byte-pair encoding[17] (BPE) (Sennrich et al., 2016b) processing on the combination of in-domain bilingual data and general-domain bilingual data, and set the number of BPE merge operations to 50,000 for source and target sides, respectively.

Model Training The learning rate is set to 0.0007. All models are trained for 600K steps on 8 Tesla V100 GPUs where each is allocated with a batch size of 8192 tokens.

4.3 German-English Results

For German-English task, we first train the models on the general-domain data. Then we combine the general-domain data and the in-domain data and train the models from scratch. Finally, we introduce the synthetic bilingual data to the combination data and use all data to train the models. The model

[14] https://github.com/moses-smt/mosesdecoder/tree/master/scripts

[15] normalize-punctuation.perl, tokenizer.perl, remove-non-printing-char.perl

[16] https://github.com/moses-smt/mosesdecoder/

[17] https://github.com/rsennrich/subword-nmt

Dataset	Size	DEEP	HYBRID	BIG	LARGE	ENSEMBLE
General Domain	37.8M	37.62	37.81	38.03	38.27	38.95
+ In-domain Data	2.5M	38.18	38.12	38.65	39.56	40.22
+ BT In-Domain Data	5.4M	38.55	38.74	38.85	40.16	40.68

Table 3: BLEU scores on the WMT19 German⇒English biomedical test set. Only the correctly aligned sentences are used in the test set.

Dataset	Size	DEEP	HYBRID	BIG	LARGE	ENSEMBLE
General Domain	19.1M	20.31	19.56	19.41	20.52	21.26
+ BT In-Domain Data	5.4M	28.52	28.83	29.32	29.80	31.34
+ OOD In-house Data	8.5M	29.92	30.07	30.66	32.05	33.23

Table 4: BLEU scores on the WMT19 Chinese⇒English biomedical test set. Only the correctly aligned sentences are used in the test set.

with best validation loss throughout the training process is selected as the final model for the testing. For model inference, the length penalty is set to 0.6 and the beam size is set to 4.

The German-English results are listed in Table 3. Our observations are:

- Due to the largest model capacity, LARGE model obtains the best translation performance among the four model variants.
- Ensemble decoding with different transformer architectures (ENSEMBLE in Table 3) achieves best translation performance.
- Leveraging in-domain bilingual data ("+In-domain") and synthetic bilingual data ("+BT In-domain") achieves significant translation improvement.

Data rejuvenation[18] (Jiao et al., 2020) is an approach which exploits the inactive training examples for neural machine translation on large-scale datasets. We adopt the data rejuvenation approach to German⇒English translation task. Experimental results are presented in Tale 7 and the data rejuvenation approach achieves significant improvement over the baseline LARGE model.

4.4 Chinese-English Results

For Chinese-English task, we gradually add the general-domain data, the synthetic bilingual data and OOD in-house data to the training data and train the models from scratch. Since the development set and test set have different data distribution, we save checkpoints every epoch and average the last 5 checkpoints rather than choose the model with best validation loss. For model inference, the length penalty is set to 2.0 and the beam size is set to 8.

Similar phenomena are observed in Chinese-English translation task. Table 4 shows Chinese-English translation results. Finally, our systems obtain 32.24 BLEU points and 33.23 BLEU points on the development and test sets, respectively.

4.5 Main Results

Main results are reported in Table 5. Our submissions (**T**encent AI Lab **M**achine **T**ranslation, TMT) with model ensemble technique achieve strong performances in WMT19 German⇔English and Chinese⇔English biomedical test sets.

5 Official Results

The official automatic evaluation results of our submissions for WMT 2020 are presented in Table 6. Our final systems rank the 1st and 3rd places on German-English and English–German, respectively, in terms of BLEU score.

6 Conclusion

In this paper, we present Tencent AI Lab machine translation systems for the WMT20 biomedical translation shared task and release the pre-trained models as well as the in-domain synthetic Chinese-English bilingual data for the research commu-

[18] https://github.com/wxjiao/Data-Rejuvenation

System	De-En	En-De	Zh-En	En-Zh
ARC (Peng et al., 2019)	38.84	35.39	32.16	37.09
UCAM (Saunders et al., 2019)	38.07	34.69	n/a	n/a
Our System	40.68	35.53	33.23	37.85

Table 5: Evaluation of translation performance on the WMT19 German⇔English and Chinese⇔English biomedical test sets. Only the correctly aligned sentences are used in the test sets.

System	De-En	En-De	Zh-En	En-Zh
Best Official	41.65	36.89	35.28	46.86
TMT Primary Run	41.65	35.24	30.48	39.43

Table 6: Official BLEU scores of our submissions for WMT20 biomedical task. Only the correctly aligned sentences are used in the test sets.

	Dev	Bio19
LARGE	52.37	39.56
+*data rejuvenation*	52.69	40.31

Table 7: Effect of data rejuvenation strategy. BLEU scores on the WMT19 German⇒English biomedical test set. Only the correctly aligned sentences are used in the test set.

nity. Our systems in German-English and English-German are ranked 1st and the 3rd respectively according to the official evaluation results in terms of BLEU scores. We also participate in the news translation (Wu et al., 2020) and the chat translation tasks (Wang et al., 2020).

In the future, we plan to explore domain adaptation (Peng et al., 2019; Saunders et al., 2019; Chu and Wang, 2018; Wang et al., 2017a), phrase modeling (Wang et al., 2017b,c; Hao et al., 2019a), structural modeling (Hao et al., 2019c; Wang et al., 2019b) strategies to improve the system performance.

7 Acknowledgments

We thank Yongchang Hao for his implementation of Hybrid TRANSFORMER model, Wenxiang Jiao for his implementation of DATA REJUVENATION, and the anonymous reviewers for their insightful suggestions.

References

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. 2011. Domain adaptation via pseudo in-domain data selection. In *EMNLP*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR*.

Rachel Bawden, Kevin Bretonnel Cohen, Cristian Grozea, Antonio Jimeno Yepes, Madeleine Kittner, Martin Krallinger, Nancy Mah, Aurélie Névéol, Mariana Neves, Felipe Soares, et al. 2019. Findings of the wmt 2019 biomedical translation shared task: Evaluation for medline abstracts and biomedical terminologies. In *WMT*.

Chenhui Chu and Rui Wang. 2018. A survey of domain adaptation for neural machine translation. In *COLING*.

Zi-Yi Dou, Zhaopeng Tu, Xing Wang, Shuming Shi, and Tong Zhang. 2018. Exploiting deep representations for neural machine translation. In *EMNLP*.

Zi-Yi Dou, Xing Wang, Shuming Shi, and Zhaopeng Tu. 2019. Exploiting deep representations for natural language processing. *Neurocomputing*.

Jie Hao, Xing Wang, Shuming Shi, Jinfeng Zhang, and Zhaopeng Tu. 2019a. Multi-granularity self-attention for neural machine translation. In *EMNLP-IJCNLP*, pages 886–896.

Jie Hao, Xing Wang, Shuming Shi, Jinfeng Zhang, and Zhaopeng Tu. 2019b. Towards better modeling hierarchical structure for self-attention with ordered neurons. In *EMNLP-IJCNLP*, pages 1336–1341.

Jie Hao, Xing Wang, Baosong Yang, Longyue Wang, Jinfeng Zhang, and Zhaopeng Tu. 2019c. Modeling recurrence for transformer. In *NAACL*.

Wenxiang Jiao, Xing Wang, Shilin He, Irwin King, Michael Lyu, and Zhaopeng Tu. 2020. Data rejuvenation: Exploiting inactive training examples for neural machine translation. In *EMNLP*.

Huda Khayrallah and Philipp Koehn. 2018. On the impact of various types of noise on neural machine translation. In *WMT*.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *WMT*.

Bei Li, Yinqiao Li, Chen Xu, Ye Lin, Jiqiang Liu, Hui Liu, Ziyang Wang, Yuhao Zhang, Nuo Xu, Zeyang Wang, et al. 2019. The niutrans machine translation systems for wmt19. In *WMT*.

Le Liu, Yu Hong, Hao Liu, Xing Wang, and Jianmin Yao. 2014. Effective selection of translation model training data. In *ACL*.

Yuchen Liu, Long Zhou, Yining Wang, Yang Zhao, Jiajun Zhang, and Chengqing Zong. 2018. A comparable study on model averaging, ensembling and reranking in nmt. In *CCF International Conference on Natural Language Processing and Chinese Computing*.

Myle Ott, Michael Auli, David Grangier, and Marc'Aurelio Ranzato. 2018. Analyzing uncertainty in neural machine translation. In *ICML*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. Fairseq: A fast, extensible toolkit for sequence modeling. *NAACL*.

Wei Peng, Jianfeng Liu, PRC Shenzhen, Liangyou Li, and Qun Liu. 2019. Huawei's nmt systems for the wmt 2019 biomedical translation task. *WMT*.

Danielle Saunders, Felix Stahlberg, and Bill Byrne. 2019. Ucam biomedical translation at wmt19: Transfer learning multi-domain ensembles. In *WMT*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data. In *ACL*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *ACL*.

Yikang Shen, Shawn Tan, Alessandro Sordoni, and Aaron Courville. 2019. Ordered neurons: Integrating tree structures into recurrent neural networks. In *ICLR*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*.

Longyue Wang, Zhaopeng Tu, Xing Wang, Li Ding, Liang Ding, and Shuming Shi. 2020. Tencent AI Lab machine translation systems for the WMT20 chat translation task. In *Proceedings of the Fifth Conference on Machine Translation*.

Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F Wong, and Lidia S Chao. 2019a. Learning deep transformer models for machine translation. In *ACL*.

Rui Wang, Masao Utiyama, Lemao Liu, Kehai Chen, and Eiichiro Sumita. 2017a. Instance weighting for neural machine translation domain adaptation. In *EMNLP*.

Xing Wang, Zhengdong Lu, Zhaopeng Tu, Hang Li, Deyi Xiong, and Min Zhang. 2017b. Neural machine translation advised by statistical machine translation. In *AAAI*.

Xing Wang, Zhaopeng Tu, Longyue Wang, and Shuming Shi. 2019b. Self-attention with structural position representations. In *EMNLP-IJCNLP*.

Xing Wang, Zhaopeng Tu, Deyi Xiong, and Min Zhang. 2017c. Translating phrases in neural machine translation. In *EMNLP*.

Shuangzhi Wu, Xing Wang, Longyue Wang, Fangxu Liu, Jun Xie, Zhaopeng Tu, Shuming Shi, and Mu Li. 2020. Tencent neural machine translation systems for the WMT20 news translation task. In *Proceedings of the Fifth Conference on Machine Translation*.

ParBLEU: Augmenting Metrics with Automatic Paraphrases for the WMT'20 Metrics Shared Task

Rachel Bawden[1] **Biao Zhang**[1] **Andre Tättar**[2] **Matt Post**[3]
[1]School of Informatics, University of Edinburgh, Scotland
[2]University of Tartu, Tartu, Estonia
[3]Johns Hopkins University, Baltimore, Maryland, USA

Abstract

We describe parBLEU, parCHRF++, and parESIM, which augment baseline metrics with automatically generated paraphrases produced by PRISM (Thompson and Post, 2020a), a multilingual neural machine translation system. We build on recent work studying how to improve BLEU by using diverse automatically paraphrased references (Bawden et al., 2020), extending experiments to the multilingual setting for the WMT2020 metrics shared task and for three base metrics. We compare their capacity to exploit up to 100 additional synthetic references. We find that gains are possible when using additional, automatically paraphrased references, although they are not systematic. However, segment-level correlations, particularly into English, are improved for all three metrics and even with higher numbers of paraphrased references.

1 Introduction

One of the major challenges faced when automatically evaluating machine translation (MT) outputs is that there are almost always multiple correct translations of a sentence, and an automatic metric should be able to reward them all. Some of the most widely used MT metrics, including BLEU (Papineni et al., 2002) and CHRF++ (Popović, 2015, 2017), rely on a surface-form comparison of MT outputs to a human-produced reference translation. Both metrics support the use of multiple references. However, even for metrics that support multiple references, human-produced references are expensive to produce and so are rarely available. To overcome this problem, metrics that do not rely on the surface form of reference translations have been developed. One example is ESIM (Chen et al., 2017; Mathur et al., 2019), which uses contextual embeddings with the aim of creating an abstract meaning representation of the reference, with the potential of covering all translations with the correct meaning.

We explore an alternative way of increasing the capacity of MT metrics to reward multiple valid translations: create additional references by automatically paraphrasing the original reference. There have been previous efforts to provide some sort of paraphrase support, mostly concentrating on synonyms (Banerjee and Lavie, 2005; Kauchak and Barzilay, 2006; Denkowski and Lavie, 2014). However, we base our work on a more recent attempt to improve BLEU using diverse automatic parahrasing with high quality MT-style *sentential* paraphrasing (Bawden et al., 2020).

We put this to the test in the WMT'20 metrics shared task by applying Bawden et al.'s (2020) approach to three different metrics: BLEU, CHRF++ and ESIM. We compare the different metrics' capacity to exploit automatically generated multiple references. We choose to use diverse paraphrases produced using PRISM (Thompson and Post, 2020a), since they are available in multiple languages, including most languages of the WMT shared task. We find that gains in correlation are possible, but this depends largely on the language direction and on whether the metric is system- or segment-level. The most positive gains are seen at the segment level, especially for into-English and even at higher numbers of additional paraphrases. This holds for all three metrics, despite ESIM relying on more abstract semantic representations.

2 Overview of Base Metrics

In an extension of (Bawden et al., 2020), we augment three base metrics with automatic paraphrasing. The metrics vary in the basic units of comparison between MT outputs and the reference. BLEU and CHRF++ compare surface representations, BLEU at the token level, whereas CHRF++ also takes into account character n-grams. ESIM is an embedding-based metric, which aims to cap-

Proceedings of the 5th Conference on Machine Translation (WMT), pages 887–894
Online, November 19–20, 2020.

ture the semantic relatedness of the sentences. A description of each base metric can be found below.

2.1 BLEU

BLEU (Papineni et al., 2002) is the dominant metric in MT. It is a modified form of n-gram precision, calculated by averaging token n-gram precisions (p_n, $n = 1..4$) and multiplying by a brevity penalty (BP) used to penalise overly short translations:

$$\text{BLEU} = \text{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right) \quad (1)$$

$$\text{BP} = \begin{cases} 1 & \text{if } c > r \\ e^{1-r/c} & \text{if } c \leq r \end{cases} \quad (2)$$

$$p_n = \frac{\sum_{h \in H} \sum_{\text{ngram} \in h} \#_{\text{clip}}\left(ngram\right)}{\sum_{h' \in H} \sum_{\text{ngram'} \in h'} \#\left(ngram'\right)}, \quad (3)$$

where c and r are the lengths of the hypothesis and reference sets respectively, H is the set of hypothesis translations, $\#\left(ngram\right)$ the number of times *ngram* appears in the hypothesis, and $\#_{\text{clip}}(ngram)$ is the same but clipped to the maximum number of times it appears in any one reference (if several references are available).

BLEU is typically used in its corpus-based variant, where a single score is produced for a test set. However, a segment-level variant also exists, where each sentence is scored individually. Smoothing is necessary in this segment-level variant to counteract the effect of 0 n-gram precision.

We use the sacreBLEU implementation[1] of BLEU (Post, 2018), with default tokenisation (and `-tok zh` tokenisation for Chinese) and exponential smoothing for the sentence-level variant.

2.2 CHRF++

CHRF++ (Popović, 2017),[2] like BLEU, is a surface-based metric, but which relies on overlap in character n-grams as well as token n-grams (hence its name 'character n-gram F-score'). This theoretically gives it an advantage over BLEU, since it is able to reward partial token matches thanks to its character-level component.

The original chrF (Popović, 2015) was calculated as follows using just character-level n-grams:

$$ngrF\beta = (1 + \beta^2)\frac{ngrP \cdot ngrR}{\beta^2 \cdot ngrP + ngrR},$$

where $ngrP$ and $ngrR$ respectively stand for the arithmetic average of n-gram precision and recall over character n-grams from 1 to N, where β gives more or less weight to precision than recall. CHRF++ expands on this original metric by also including token-level n-grams in this calculation with n from 1 to M. The best results were found with $N = 6$ and $M = 1$ or 2. We use the settings used in the WMT19 shared task: $N = 6$, $M = 1$ and $\beta = 3$. Like BLEU, CHRF++ has a specific corpus-level and sentence-level variant.

2.3 ESIM

ESIM (Chen et al., 2017; Mathur et al., 2019), is an embedding-based metric, which relies on neural models to handle inter-sentence semantic relatedness, going beyond surface-level matching (as in BLEU and CHRF++). ESIM was originally proposed to compare and match sentence pairs for natural language inference (Chen et al., 2017). Mathur et al. (2019) adapted it to evaluate MT performance by pairing the human reference and the MT output as ESIM input. Following (Mathur et al., 2019),[3] we treat the evaluation task as a regression task, and train ESIM models on segment-level human judgments. We train ESIM on the WMT18 metric data for WMT19 evaluation, and WMT18+WMT19 metric data for WMT20 evaluation. ESIM is a sentence-level metric. Scores are averaged to produce a single score for a given corpus.

3 Experiment Setup

Paraphrase generation We use the PRISM system to generate paraphrases. PRISM is a many-many multilingual NMT system covering 39 languages, including all of those of WMT 2019, except Gujarati. In their submission to the WMT 2020 Metrics task, Thompson and Post (2020c) retrained PRISM with five additional languages: Gujarati (for WMT'19), and Inuktitut, Khmer, Pashto, and Tamil (for WMT'20). This provided almost complete coverage of the WMT 2019 and 2020 languages. We use this same model.

By design, PRISM approaches paraphrasing as a zero-shot translation task. As a result, while good for scoring, it is not a particularly good generative model, in terms of being able to produce diverse outputs. Thompson and Post (2020b) have tried to address this, but their implementation does not

[1] https://github.com/mjpost/sacrebleu
[2] https://github.com/m-popovic/chrF/
[3] https://github.com/nitikam/mteval-in-context

We reached a pretty quick agreement, Kouki said.
We reached a fairly quick agreement, Kouki said.
We reached a fairly quick agreement, Kouki was quoted as saying.
We reached a fairly quick agreement, Kouki said
We reached a fairly quick agreement, Kouki was quoted as telling reporters.
We reached a fairly quick agreement, Kouki was quoted to be quoted as saying.
We reached a fairly quick agreement, Kouki was quoted as adding.

Jamsen says the church bells don't ring because of a malfunction.
Jamsen says the church bells do not ring because of a malfunction.
Jamsen says the church bells do not ring because of a maloperation.
Jamsen says the church bells aren't ringing because of an improper functioning.
Jamsen says that the church bells don't ring because of a mal-function.
It's a technical malfunctory, to say it's a technical malfunction.
It's a technical malfunction, I'm sure.

Figure 1: Two examples of automatic paraphrasing from fi–en WMT'20 (original references in bold).

produce n-best lists. We therefore produce n-best lists from the model using Fairseq's built-in diverse beam search tool. For every reference in the WMT19 and WMT20 news test sets, we generate a 100-best list (Vijayakumar et al., 2016).[4]

Figure 1 shows examples of paraphrases of two fi–en WMT'20 references. Note that the paraphrases are divere and generally of high quality. However, the later paraphrases may be noisier.

Integrating multiple references We augment each of the base metrics described in Section 2 to produce three new metrics: parBLEU, parCHRF++ and parESIM. Both BLEU and CHRF++ have inbuilt support for multiple references. For ESIM, we calculate the score for each reference separately and then average them to get the final score.

Metrics Task Setup Awaiting the gold judgments for WMT'20, we test and report the results of each method on the WMT19 metrics task.[5] We follow the metrics task setup (Ma et al., 2019) by calculating the correlation with manual direct assessments (DA) of MT quality (Graham et al., 2013). System-level scores are evaluated using Pearson's r and segment-level correlations using Kendall's τ on the DA assessments converted into relative rankings. Statistically significant improvements (over the single-reference base metric) are marked in bold (with $p \leq 0.05$). Significance is calculated using the Williams test (Williams, 1959) at the system level and bootstrap resampling at the segment level.

[4]We pass the following arguments: `fairseq-interactive ... --beam 100 --nbest 100 --diverse-beam-groups 10 --diverse-beam-strength 1`

[5]http://statmt.org/wmt19/results.html

4 Results

The results are reported in the following three sections for each paraphrase-augmented metric: parBLEU (Section 4.1), parCHRF++ (Section 4.2) and parESIM (Section 4.3). We report results for up to 100 additional paraphrased references, except for parESIM, where we report up to 50 additional references due to the length of time needed to calculate results. There are some general trends:

- There is often a difference between into- and from-English language directions, with more positive results seen into English. This could be due to the potential better quality of the English paraphrases.
- Results are better for into-English at the segment-level, where adding paraphrases tends to help even with more paraphrases.

4.1 parBLEU

Results for parBLEU are found in Table 1 (system-level) and Table 2 (segment-level).

#extra	en-cs	en-de	en-fi	en-gu	en-kk	en-lt	en-ru	en-zh
0	0.988	0.959	0.970	0.736	0.849	0.989	0.968	0.901
1	0.986	0.954	0.968	0.737	0.876	0.982	0.977	0.941
2	0.986	0.953	0.968	0.738	0.875	0.981	0.979	0.938
5	0.986	0.954	0.968	0.738	0.879	0.980	0.980	0.933
25	0.984	0.958	0.969	0.739	0.883	0.976	0.982	0.927
50	0.982	0.959	0.969	0.740	0.887	0.974	0.982	0.924
100	0.977	0.957	0.965	0.743	0.888	0.973	0.982	0.897

(a) From-English language directions

#extra	de-en	fi-en	gu-en	kk-en	lt-en	ru-en	zh-en
0	0.891	0.986	0.798	0.943	0.969	0.861	0.888
1	**0.905**	0.987	0.802	0.951	**0.975**	**0.887**	**0.898**
2	**0.906**	0.987	0.797	0.953	**0.977**	**0.893**	0.894
5	**0.912**	0.987	0.794	**0.955**	**0.981**	**0.897**	0.892
25	**0.925**	0.985	0.784	**0.966**	**0.984**	**0.898**	0.894
50	**0.930**	0.984	0.780	**0.971**	**0.986**	**0.906**	0.892
100	**0.940**	0.979	0.777	**0.977**	**0.990**	**0.919**	0.874

(b) To-English language directions

Table 1: parBLEU system-level results.

System-level results are variable, with a notable difference between into-English and from-English language directions. For a couple of from-English languages, there are some slightly higher correlations but these are not significant, and some deteriorations can be seen when adding paraphrase for others. Adding paraphrased references is more successful for into-English languages. For four of the language directions, adding the maximum number of 100 paraphrases provides the greatest significant correlation gains, suggesting that even more gains could be achieved with more paraphrases. These gains are illustrated in Figure 2a.

#extra	en-cs	en-de	en-fi	en-gu	en-kk	en-lt	en-ru	en-zh
0	0.351	0.239	0.381	0.436	0.362	0.309	0.462	0.262
1	**0.359**	**0.259**	**0.409**	0.428	**0.370**	**0.322**	**0.483**	-0.313
2	**0.361**	**0.260**	**0.412**	0.426	0.370	**0.318**	**0.485**	-0.309
5	**0.360**	**0.260**	**0.416**	0.422	0.365	0.311	**0.487**	-0.309
25	**0.366**	**0.265**	**0.422**	0.415	0.362	0.314	**0.489**	0.263
50	**0.362**	**0.267**	**0.425**	0.414	0.357	0.318	**0.489**	0.266
100	**0.369**	**0.268**	**0.423**	0.398	0.333	**0.327**	**0.488**	-0.280

(a) From-English language directions

#extra	de-en	fi-en	gu-en	kk-en	lt-en	ru-en	zh-en
0	0.050	0.223	0.166	0.363	0.248	0.106	0.312
1	**0.055**	**0.227**	**0.175**	0.367	**0.264**	**0.113**	**0.321**
2	**0.054**	0.226	**0.178**	0.362	**0.268**	**0.114**	**0.320**
5	**0.056**	0.227	**0.181**	0.363	**0.272**	**0.112**	0.317
25	**0.065**	**0.235**	**0.185**	**0.372**	**0.275**	**0.126**	**0.321**
50	**0.070**	**0.241**	**0.186**	**0.375**	**0.284**	**0.127**	**0.324**
100	**0.066**	**0.243**	**0.191**	**0.371**	**0.293**	**0.134**	0.311

(b) To-English language directions

Table 2: parBLEU segment-level results.

Segment-level results show variability according to the language direction too. The greatest gains are seen for the into-English directions, and the highest scores are achieved for the higher order numbers of paraphrases. Some gains are seen for most from-English directions, even with higher numbers of paraphrases. Interestingly, the language directions that see gains at the segment level are not correlated with those that see gains at the system level.

4.2 parCHRF++

System-level and segment-level results can be found in Table 3 and Table 4 respectively. The CHRF++ baseline (0 extra references) is higher than the BLEU baseline for into-English at the system-level and into all languages (except Chinese) at the segment-level.

#extra	en-cs	en-de	en-fi	en-gu	en-kk	en-lt	en-ru	en-zh
0	0.984	0.977	0.981	0.836	0.967	0.969	0.985	0.801
1	0.981	0.977	0.979	0.836	0.972	0.967	0.986	0.821
2	0.980	0.977	0.978	0.835	0.972	0.966	0.986	0.824
5	0.980	0.977	0.977	0.835	0.973	0.965	0.986	0.827
10	0.979	0.977	0.977	0.835	0.973	0.965	0.986	0.827
25	0.979	0.976	0.976	0.835	0.974	0.964	0.986	0.823
50	0.979	0.976	0.975	0.835	0.974	0.963	0.985	0.821
100	0.974	0.976	0.972	0.835	0.973	0.962	0.986	0.823

(a) From-English language directions

#extra	de-en	fi-en	gu-en	kk-en	lt-en	ru-en	zh-en
0	0.909	0.991	0.947	0.966	0.936	0.918	0.955
1	**0.919**	0.991	0.948	0.967	**0.948**	**0.930**	**0.961**
2	**0.922**	0.991	0.948	0.968	**0.950**	**0.932**	**0.960**
5	**0.925**	0.991	0.948	0.969	**0.952**	**0.936**	**0.961**
25	**0.930**	0.991	0.952	**0.972**	**0.952**	**0.940**	**0.962**
50	**0.933**	0.991	0.953	**0.973**	**0.953**	**0.942**	**0.962**
100	**0.938**	0.990	0.950	**0.982**	**0.963**	**0.949**	0.963

(b) To-English language directions

Table 3: parCHRF++ system-level results.

At the system level, as with parBLEU, greater gains are seen for into-English than from-English language directions: all into-English language directions bar fi–en show increases. Moreover, most into-English language directions continue to see improvements with higher numbers of references. This trend can be seen in Figure 2b.

#extra	en-cs	en-de	en-fi	en-gu	en-kk	en-lt	en-ru	en-zh
0	0.449	0.323	0.518	0.546	0.497	0.439	0.548	0.238
1	**0.455**	**0.326**	0.519	0.546	0.498	0.431	**0.564**	0.237
5	0.448	0.325	0.517	0.546	0.490	0.418	**0.555**	0.221
25	0.444	**0.327**	0.515	0.545	0.487	0.416	**0.560**	0.209
50	0.443	**0.327**	0.515	0.545	0.485	0.417	**0.559**	0.203
100	0.433	0.322	0.506	0.545	0.459	0.405	0.546	0.193

(a) From-English language directions

#extra	de-en	fi-en	gu-en	kk-en	lt-en	ru-en	zh-en
0	0.125	0.288	0.254	0.393	0.303	0.182	0.373
1	0.123	0.288	**0.258**	0.396	**0.311**	0.182	0.375
5	0.126	0.286	**0.261**	**0.398**	**0.317**	0.182	0.377
25	**0.129**	0.291	**0.263**	**0.398**	**0.322**	0.183	0.375
50	0.128	0.291	**0.265**	0.397	**0.327**	0.182	0.378
100	0.120	0.285	**0.269**	0.397	**0.313**	0.180	0.367

(b) To-English language directions

Table 4: parCHRF++ segment-level results.

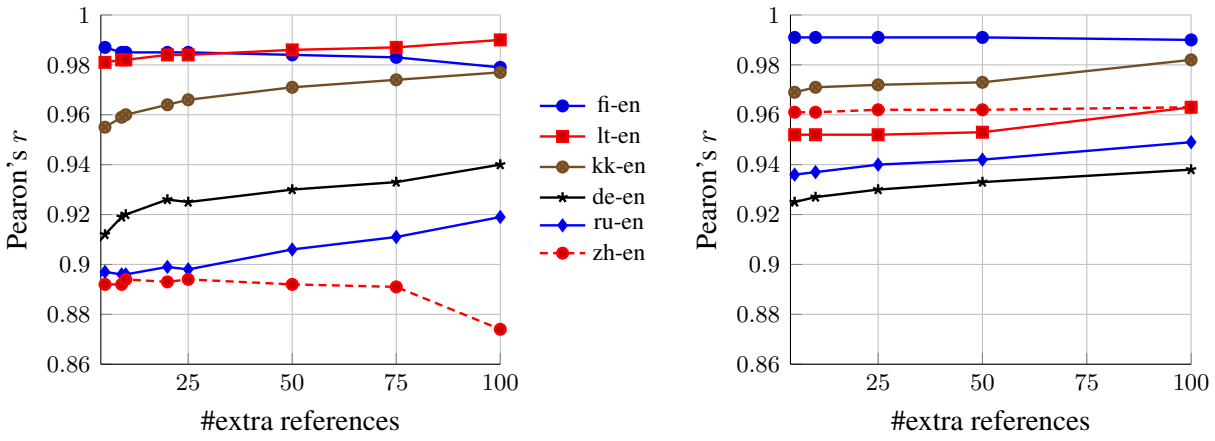
At the segment level, extra references help all into-English directions, although this does depend on the number of references added for some language directions. From-English, some slight gains are seen but in most cases, adding extra references degrades results. At the segment level, the best results can be seen with just one additional reference.

4.3 parESIM

System-level and segment-level results can be found in Table 5 and Table 6 respectively. As an automatic metric that relies on comparing continuous representations (aiming to abstract away from surface forms), we would expect paraphrases to help ESIM less than the two other metrics, for which surface form variation is one of the major limitations.

At the system level, additional paraphrases does not seem to help for any of the language directions, and is even harmful (decreasing correlations as the number of paraphrases is increased). This could be due to the addition of noise in the results, which treats semantically divergent hypotheses as valid. Note however that the correlations start from a strong base—baseline ESIM has a much higher correlation than BLEU and CHRF++.

The segment-level results are more positive: paraphrasing significantly helps four from-English directions (into cs, de, ru and zh). It brings even more positive gains for the into-English language directions, where the best results are often achieved with the higher numbers of additional paraphrases.



(a) parBLEU (b) parCHRF++

Figure 2: System-level results for into-English language directions.

#extra	en-cs	en-de	en-fi	en-gu	en-kk	en-lt	en-ru	en-zh
0	0.924	0.990	0.943	0.946	0.978	0.960	0.978	0.942
1	0.918	0.988	0.935	0.942	0.972	0.941	0.977	0.947
2	0.916	0.987	0.932	0.935	0.970	0.932	0.977	0.949
5	0.913	0.986	0.928	0.916	0.967	0.921	0.975	0.949
10	0.912	0.985	0.926	0.895	0.966	0.916	0.975	0.949
25	0.910	0.985	0.924	0.870	0.966	0.912	0.973	0.949
50	0.909	0.984	0.923	0.857	0.966	0.910	0.973	0.950

(a) From-English language directions

#extra	de-en	fi-en	gu-en	kk-en	lt-en	ru-en	zh-en
0	0.938	0.967	0.875	0.982	0.987	0.973	0.988
1	0.937	0.967	0.873	0.980	0.987	0.972	0.989
2	0.937	0.967	0.873	0.980	0.987	0.971	0.989
5	0.936	0.966	0.872	0.978	0.987	0.971	0.989
10	0.935	0.966	0.872	0.976	0.987	0.971	0.989
25	0.935	0.965	0.871	0.974	0.987	0.970	0.989
50	0.934	0.965	0.870	0.972	0.987	0.969	0.989

(b) To-English language directions

Table 5: parESIM system-level results.

#extra	en-cs	en-de	en-fi	en-gu	en-kk	en-lt	en-ru	en-zh
0	0.471	0.356	0.535	0.545	0.508	0.487	0.582	0.330
1	0.475	**0.366**	0.532	0.517	0.494	0.489	**0.593**	**0.352**
2	**0.476**	**0.364**	0.526	0.488	0.476	0.477	**0.593**	**0.348**
5	**0.480**	**0.368**	0.520	0.413	0.452	0.467	**0.596**	**0.344**
10	**0.477**	**0.370**	0.519	0.359	0.436	0.467	**0.595**	**0.345**
25	0.475	**0.370**	0.514	0.307	0.426	0.463	**0.589**	**0.339**
50	0.476	**0.370**	0.515	0.290	0.424	0.464	**0.592**	**0.343**

(a) From-English language directions

#extra	de-en	fi-en	gu-en	kk-en	lt-en	ru-en	zh-en
0	0.155	0.328	0.294	0.426	0.348	0.190	0.347
1	**0.163**	0.330	0.293	0.424	0.352	0.193	**0.352**
2	**0.168**	0.329	0.293	0.427	**0.353**	0.192	**0.354**
5	**0.174**	0.328	0.294	0.421	**0.354**	**0.199**	**0.355**
10	**0.174**	**0.334**	0.294	0.419	**0.356**	**0.199**	**0.354**
25	**0.175**	**0.334**	0.295	0.415	**0.359**	**0.201**	**0.354**
50	**0.176**	**0.333**	0.294	0.412	**0.357**	**0.200**	**0.356**

(b) To-English language directions

Table 6: parESIM segment-level results.

4.4 Additional parBLEU comparisons

Following the shared task, we explored some alternative versions of parBLEU.

Replacing the original reference Concurrently to Bawden et al. (2020), Freitag et al. (2020) also review paraphrasing for BLEU, although they focus on human paraphrasing. They find that better correlations are achieved by replacing the original reference with a human paraphrased one, as original references often display translationese. We test this observation here, but using our automatic paraphrases. Results are shown in Table 7 (system level) and Table 8 (segment level).

#extra	en-cs	en-de	en-fi	en-gu	en-kk	en-lt	en-ru	en-zh
Original	0.988	0.959	0.970	0.736	0.849	0.989	0.968	0.901
Paraphrased	0.978	0.946	0.951	0.115	**0.941**	0.946	0.983	0.936

(a) From-English language directions

Metric	de-en	fi-en	gu-en	kk-en	lt-en	ru-en	zh-en
Original	0.891	0.986	0.798	0.943	0.969	0.861	0.888
Paraphrased	**0.916**	0.988	0.799	0.952	**0.978**	**0.905**	0.902

(b) To-English language directions

Table 7: parBLEU system-level results when using the original reference versus the first paraphrase.

#extra	en-cs	en-de	en-fi	en-gu	en-kk	en-lt	en-ru	en-zh
Original	0.351	0.239	0.381	0.436	0.362	0.309	0.462	0.262
Paraphrased	0.327	0.228	0.342	-0.149	0.224	0.181	0.455	0.210

(a) From-English language directions

Metric	de-en	fi-en	gu-en	kk-en	lt-en	ru-en	zh-en
Original	0.050	0.223	0.166	0.363	0.248	0.106	0.312
Paraphrased	0.046	0.222	0.167	0.360	0.253	0.106	0.316

(b) To-English language directions

Table 8: parBLEU segment-level results when using the original reference versus the first paraphrase.

We find that replacing the original reference with its first paraphrase results in higher correlations for the into-English language directions at the system

level (although the gain is only significant for three directions), and there do not seem to be gains at the segment level. In general, it harms both correlation types for the from-English language directions, probably due to the better quality of the English paraphrasing compared to that of the other languages. This appears to confirm Freitag et al.'s observation, as long as the quality of the paraphraser is good enough, which is our hypothesis concerning the into-English language directions.

Type of paraphraser We compare three different paraphrasers for the into-English language directions: (i) the 'sampled' diverse paraphrasing approach from (Bawden et al., 2020), (ii) the n-best PRISM paraphrases, and (iii) the n-best diverse PRISM paraphrases used elsewhere in this paper. The results are given in Table 9. Somewhat surprisingly, even though they are not designed to be diverse, the n-best paraphrases give good correlations, at least up to 20 paraphrases, which was the maximum number tested with the sampled paraphraser. The sampled paraphrases also often perform better than the diverse approach produced by the PRISM paraphraser. One reason for this could be that the sampled paraphraser is trained specifically as an English paraphraser, whereas the PRISM paraphraser is multilingual (therefore providing greater support for automatic evaluation).

Exclusion of outliers Mathur et al. (2020) suggested that system-level correlations computed with Pearson's are artificially inflated due to the presence of outliers, which are typically very poorly performing systems with low human scores. They propose a method based on *mean average deviation* (MAD) to exclude those outliers. We applied this method to the WMT19 system-level data to exclude systems, and then recomputed the system-level correlations.

The complete results are in Table 10. Comparing this to Table 7, we see an absolute drop in values, but little to nothing in the way of reversals between the BLEU (single-reference, zero-paraphrase) baseline and the paraphrase methods.

5 Conclusions and Future Work

The goal with any metric is to balance accuracy with ease-of-use. For our submission to the WMT20 metrics task, we extended our work investigating paraphrased English references (Bawden et al., 2020), by using a multilingual paraphraser.

Type	#extra	de-en	fi-en	gu-en	kk-en	lt-en	ru-en	zh-en
	0	0.891	0.986	0.798	0.943	0.969	0.861	0.888
Sampled	1	**0.915**	0.985	0.801	0.960	**0.984**	**0.907**	0.891
	2	**0.926**	0.986	0.799	0.962	**0.987**	**0.918**	0.896
	10	**0.942**	0.980	0.800	0.970	**0.992**	**0.932**	0.906
	20	**0.946**	0.976	0.800	0.973	**0.992**	0.933	0.907
n-best	1	**0.910**	0.987	0.801	0.952	**0.975**	**0.884**	**0.899**
	2	**0.913**	0.988	0.802	0.954	**0.975**	**0.894**	**0.901**
	10	**0.935**	0.989	0.801	0.959	**0.978**	**0.915**	**0.907**
	20	**0.938**	0.989	0.800	0.960	**0.981**	**0.923**	**0.913**
diverse	1	**0.905**	0.987	0.802	0.951	**0.975**	**0.887**	**0.898**
	2	**0.906**	0.987	0.797	0.953	**0.977**	**0.893**	0.894
	10	**0.061**	0.225	**0.182**	0.369	**0.272**	**0.121**	**0.320**
	20	**0.926**	0.985	0.784	**0.964**	**0.984**	**0.899**	0.893

(a) System-level

Type	#extra	de-en	fi-en	gu-en	kk-en	lt-en	ru-en	zh-en
	0	0.050	0.223	0.166	0.363	0.248	0.106	0.312
Sampled	1	**0.054**	**0.237**	**0.181**	0.364	**0.282**	**0.121**	0.309
	2	**0.057**	**0.239**	**0.185**	0.367	**0.283**	**0.119**	0.307
	10	**0.078**	**0.254**	**0.191**	**0.376**	**0.302**	**0.127**	0.314
	20	**0.077**	**0.252**	**0.192**	**0.378**	**0.308**	**0.125**	0.316
n-best	1	**0.056**	**0.227**	**0.175**	0.367	**0.261**	**0.111**	**0.324**
	2	**0.054**	0.226	**0.180**	**0.370**	**0.272**	**0.119**	**0.323**
	10	**0.064**	**0.232**	**0.190**	**0.381**	**0.278**	**0.133**	**0.324**
	20	**0.062**	**0.240**	**0.193**	**0.384**	**0.289**	**0.131**	**0.332**
diverse	1	**0.055**	**0.227**	**0.175**	0.367	**0.264**	**0.113**	**0.321**
	2	**0.054**	0.226	**0.178**	0.362	**0.268**	**0.114**	**0.320**
	10	**0.061**	0.225	**0.182**	0.369	**0.272**	**0.121**	**0.320**
	20	**0.063**	**0.234**	**0.183**	**0.371**	**0.273**	**0.124**	**0.323**

(b) Segment-level

Table 9: Correlation results for parBLEU for into-English language directions.

#extra	en-cs	en-de	en-fi	en-gu	en-kk	en-lt	en-ru	en-zh
0	0.988	0.828	0.961	0.736	0.591	0.989	0.946	0.901
1	0.986	0.827	0.953	0.737	0.560	0.982	0.964	0.941
2	0.986	0.824	0.953	0.738	0.559	0.981	0.969	0.938
5	0.986	0.826	0.951	0.738	0.563	0.980	0.972	0.933
25	0.984	0.837	0.951	0.739	0.559	0.976	0.972	0.927
50	0.982	0.837	0.948	0.740	0.563	0.974	0.972	0.924
100	0.977	0.821	0.939	0.743	0.530	0.973	0.970	0.897

(a) From-English language directions

#extra	de-en	fi-en	gu-en	kk-en	lt-en	ru-en	zh-en
0	0.828	0.986	0.967	0.917	0.968	0.844	0.823
1	0.844	0.987	0.970	0.911	**0.975**	**0.876**	0.837
2	0.843	0.987	0.971	0.913	**0.978**	**0.881**	0.827
5	0.851	0.987	0.969	0.910	**0.981**	**0.886**	0.817
25	**0.872**	0.985	0.968	0.925	**0.985**	**0.889**	0.824
50	**0.879**	0.984	0.969	0.924	**0.988**	**0.898**	0.817
100	**0.896**	0.979	0.971	0.885	**0.992**	**0.910**	0.804

(b) To-English language directions

Table 10: System-level results with parBLEU with outlier systems excluded. The *removed* row denotes how many systems were considered to be outliers

One component of ease-of-use, particularly for a metric, is to avoid highly language-specific parameter searches. Our work here used a single model and diversity parameter setting. It is possible that other approaches would yield more success: for example, varying the number of references based on reference length or complexity, or looking at other diverse generation techniques. However they are not guaranteed to work and raise questions about

the usefulness of extending surface-based metrics in the neural age. BLEU is appealing because of its simplicity and universality, but the emerging evidence (cf. Mathur et al. (2020)) suggest that the most promising approach for future work in MT evaluation is in model-based deep-learning approaches. What is encouraging and also somewhat surprising is that the embedding-based ESIM also seems to benefit from the addition of automatically paraphrased references at the segment level, especially into English.

Acknowledgements

This work was supported by funding from the European Union's Horizon 2020 research and innovation programme under grant agreements No 825299 (GoURMET), 825303 and the UK Engineering and Physical Sciences Research Council (EPSRC) fellowship grant EP/S001271/1 (MT-Stretch).

References

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Rachel Bawden, Biao Zhang, Lisa Yankovskaya, Andre Tättar, and Matt Post. 2020. A Study in Improving BLEU Reference Coverage with Diverse Automatic Paraphrasing. In *Findings of the 2020 Conference on Empirical Methods in Natural Language Processing*.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017. Enhanced LSTM for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1657–1668, Vancouver, Canada. Association for Computational Linguistics.

Michael Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 376–380, Baltimore, Maryland, USA. Association for Computational Linguistics.

Markus Freitag, David Grangier, and Isaac Caswell. 2020. BLEU might be Guilty but References are not Innocent. arXiv:2004.06063.

Yvette Graham, Timothy Baldwin, Alistair Moffat, and Justin Zobel. 2013. Continuous measurement scales in human evaluation of machine translation. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 33–41, Sofia, Bulgaria. Association for Computational Linguistics.

David Kauchak and Regina Barzilay. 2006. Paraphrasing for automatic evaluation. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 455–462, New York City, USA. Association for Computational Linguistics.

Qingsong Ma, Johnny Wei, Ondřej Bojar, and Yvette Graham. 2019. Results of the WMT19 metrics shared task: Segment-level and strong MT systems pose big challenges. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 62–90, Florence, Italy. Association for Computational Linguistics.

Nitika Mathur, Timothy Baldwin, and Trevor Cohn. 2019. Putting evaluation in context: Contextual embeddings improve machine translation evaluation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2799–2808, Florence, Italy. Association for Computational Linguistics.

Nitika Mathur, Timothy Baldwin, and Trevor Cohn. 2020. Tangled up in BLEU: Reevaluating the evaluation of automatic machine translation evaluation metrics. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4984–4997, Online. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Maja Popović. 2017. chrF++: words helping character n-grams. In *Proceedings of the Second Conference on Machine Translation*, pages 612–618, Copenhagen, Denmark. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Brian Thompson and Matt Post. 2020a. Automatic machine translation evaluation in many languages via zero-shot paraphrasing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Online. Association for Computational Linguistics.

Brian Thompson and Matt Post. 2020b. Paraphrase generation as zero-shot multilingual translation: Disentangling semantic similarity from lexical and syntactic diversity. In *Proceedings of the Fifth Conference on Machine Translation (Volume 1: Research Papers)*, Online. Association for Computational Linguistics.

Brian Thompson and Matt Post. 2020c. PRISM: The JHU Submission to the Metrics Shared Task at WMT'20. In *Proceedings of the Fifth Conference on Machine Translation (Volume 2: Shared Task Papers)*, Online. Association for Computational Linguistics.

Ashwin K. Vijayakumar, Michael Cogswell, Ramprasaath R. Selvaraju, Qing Sun, Stefan Lee, David J. Crandall, and Dhruv Batra. 2016. Diverse beam search: Decoding diverse solutions from neural sequence models. *CoRR*, abs/1610.02424.

Evan James Williams. 1959. *Regression Analysis*. Wiley, New York.

Extended Study on Using Pretrained Language Models and YiSi-1 for Machine Translation Evaluation

Chi-kiu Lo
Multilingual Text Processing
Digital Technologies Research Centre
National Research Council Canada (NRC-CNRC)
1200 Montreal Road, Ottawa, ON K1A 0R6, Canada
chikiu.lo@nrc-cnrc.gc.ca

Abstract

We present an extended study on using pretrained language models and YiSi-1 for machine translation evaluation. Although the recently proposed contextual embedding based metrics, YiSi-1, significantly outperform BLEU and other metrics in correlating with human judgment on translation quality, we have yet to understand the full strength of using pretrained language models for machine translation evaluation. In this paper, we study YiSi-1's correlation with human translation quality judgment by varying three major attributes (which architecture; which intermediate layer; whether it is monolingual or multilingual) of the pretrained language models. Results of the study show further improvements over YiSi-1 on the WMT 2019 Metrics shared task. We also describe the pretrained language model we trained for evaluating Inuktitut machine translation output.

1 Introduction

Recent research on large-scale evaluation of automatic machine translation (MT) evaluation metrics (Ma et al., 2018, 2019; Mathur et al., 2020) showed that the newly proposed contextual embedding based metrics, like YiSi-1, BERTscore (Zhang et al., 2020) and ESIM (Mathur et al., 2019), significantly outperform BLEU (Papineni et al., 2002) and other metrics in correlating with human judgment on translation quality. YiSi-1 and BERTscore use contextual embeddings extracted from the pretrained language model, Devlin et al. (2018), as-is without further fine-tuning or fitting to existing labeled data predictions. Although fine-tuning the pretrained language models for specific downstream tasks show improvements in many cases, using the pretrained language models without fine-tuning makes the MT evaluation metrics more portable to languages without labeled data and the resulted metric scores are comparable to each other over time across systems. Thus, instead of spending efforts into fine-tuning the pretrained language models for MT evaluation, we focus on finding the most optimal way (which architecture; which intermediate layer; whether it is a monolingual or multilingual model) to utilize them as-is.

Zhang et al. (2020) investigated into a few aspects (architecture and layer) of the use of contextual embeddings in text generation evaluation. They evaluated several model architectures of different sizes, such as BERT (Devlin et al., 2018), RoBERTa (Liu et al., 2019), XLNet (Yang et al., 2019) and XLM (Lample and Conneau, 2019). As more pretrained language models that cover more languages are released since then, we extend the study on YiSi-1 to include more pretrained language models and also compare the effect of using monolingual pretrained models versus multilingual pretrained models.

In this paper, we experiment on different settings of YiSi-1 in the WMT 2019 metrics shared task, integrating it with different transformer-based (Vaswani et al., 2017) contextual language models in both monolingual or multilingual, such as BERT (Devlin et al., 2018), ALBERT (Lan et al., 2020), BART (Lewis et al., 2019), RoBERTa (Liu et al., 2019), XLM-RoBERTa (Conneau et al., 2020) and XLNET (Yang et al., 2019), using different intermediate layers. We show that YiSi-1's correlation with human judgment on translation quality is improved by using the results of this study.

2 YiSi

YiSi (Lo, 2019) is a unified semantic MT quality evaluation and estimation metric for languages with different levels of available resources. YiSi-1 measures the similarity between a machine

Proceedings of the 5th Conference on Machine Translation (WMT), pages 895–902

translation and human references by aggregating weighted distributional (lexical) semantic similarities, and optionally incorporating shallow semantic structures. YiSi-0 is the degenerate version of YiSi-1 that is ready-to-deploy to any languages by using longest common character substring, instead of cosine similarity of contextual embeddings, to measure lexical similarity.

YiSi-2 is the bilingual, reference-less version, which uses bilingual word embeddings to evaluate cross-lingual lexical semantic similarity between the input and MT output, and optionally incorporating shallow semantic structures. Improvements in YiSi-2 for WMT 2020 metrics shared task is detailed in (Lo and Larkin, 2020).

2.1 Pretrained Language Models

YiSi-1 relies on a language representation to evaluate the lexical semantic similarity between the reference translation and the MT output. In WMT 2019 metrics shared task, it used pretrained BERT (Devlin et al., 2018) for this purpose.

BERT captures the sentence context in the embeddings, such that the embedding of the same subword unit in different sentences would be different from each other and be better represented in the embedding space. Monolingual BERT pretrained model for English and Chinese and multilingual BERT pretrained that covers the 104 largest languages in Wikipedia were public released in 2019.

2.1.1 Monolingual models

Monolingual BERT in other languages After the success of using monolingual BERT models for downstream NLP tasks in Chinese and English, a number of monolingual BERT models in other languages have been publicly released, such as German (Chan et al., 2019), Finnish (Virtanen et al., 2019), French (Martin et al., 2020), Japanese Inui Laboratory (2019), Dutch (de Vries et al., 2019). In our experiments, we compare the performance of YiSi-1 using these monolingual models against that using multilingual language models.

Other monolingual models in English A number of modifications to BERT have been proposed to optimize the pretrained language models. Lan et al. (2020) proposed ALBERT to reduce the amount of parameters in BERT for lower memory consumption and faster training speed. BART (Lewis et al., 2019) is effective when fine tuned for text generation tasks. RoBERTa (Liu et al., 2019) is a more robustly trained version of BERT where the key hyperparameters are empirically chosen. XLNET (Yang et al., 2019) an autoregressive model that maximizes the expected likelihood over all permutations of the input sequence factorization order. We use these models in YiSi-1 for correlation analysis with human judgment on translation quality.

2.1.2 Multilingual models

In addition to multilingual BERT used in Lo (2019), XLM-RoBERTa (Conneau et al., 2020) (XLM-R) is also a massive multilingual pretrained language model. Similar to BERT, XLM-R is also trained with a masked language model task on the concatenation of non-parallel data. The differences between XLM-R and BERT are 1) XLM-R is trained on the CommonCrawl corpus which is significantly larger than the Wikipedia training data used by BERT; 2) instead of a uniform data sampling rate used in BERT, XLM-R uses a language sampling rate that is proportional to the amount of data available in the training set. Because of these differences, XLM-R performs better on low resource languages than multilingual BERT. XLM-R covers 100 languages. In this work, we use XLM-R$_{large}$ for the best performance on lexical semantic similarity.

2.2 Inuktitut-English Cross-lingual Language Model

Since Inuktitut is not covered by any publicly released pretrained language model, we trained our own Inuktitut-English XLM (Lample and Conneau, 2019) using the Nunavut Hansard 3.0 (NH) parallel corpus (Joanis et al., 2020). The model was trained with masked language model and translation language model tasks. The Inuktitut-English XLM model has 12 layers with 8 heads and embedding size of 512.

2.3 Model size and intermediate layers

In this study, we are interested in achieving the best performance using the pretrained language models. Thus, if different sizes of the same model architecture are released, we only evaluate the largest one in out experiment. As suggested by Devlin et al. (2018); Peters et al. (2018); Zhang et al. (2020), we experimented using contextual embeddings extracted from different layers of the multilingual language encoder to find out the layer that

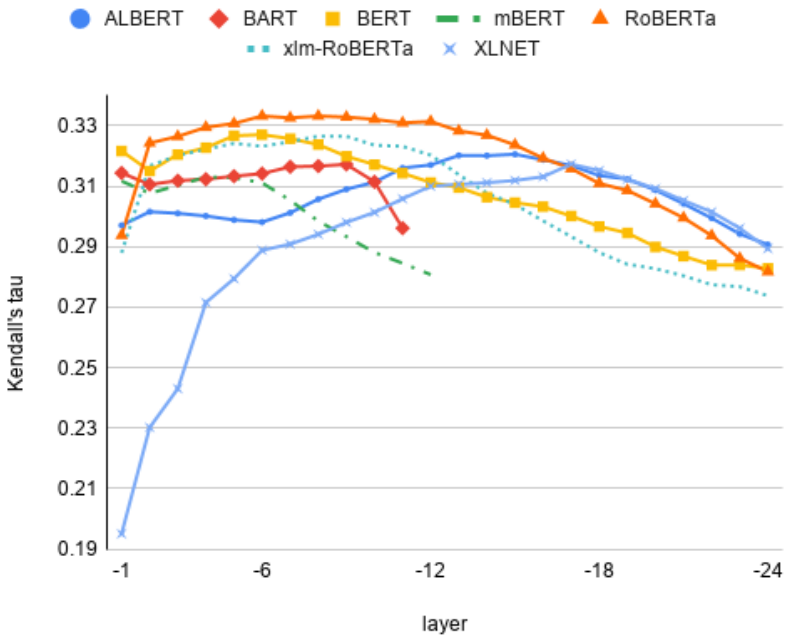


Figure 1: Average segment-level Kendall's τ correlation with human direct assessment on WMT19 *-en news translation test set of YiSi-1 using different pretrained language representation models. Solid lines represent the use of pretrained monolingual models. Dotted line represents the use of pretrained XLM-R and dashed line represents the use of pretrained multilingual BERT.

best represents the semantic space of the language.

3 Experiments and Results

We use WMT 2019 metrics shared task evaluation set (Ma et al., 2019) for our development experiments. The official human judgments for translation quality of WMT 2019 were collected using reference-based direct assessment.

Since we use exactly the same correlation analysis as the official metrics shared evaluation and the 2019 version of YiSi performed consistently well among participants in WMT 2019, we only compare our results with the 2019 version of YiSi and BLEU. Our results are directly comparable with those reported in Ma et al. (2019).

3.1 Architectures of monolingual English models

In Figure 1, we plot the change of segment-level Kendall's τ correlation of YiSi-1 across different layers of the monolingual and multilingual pretrained language models for evaluating English MT output. We see that YiSi-1 using RoBERTa$_{large}$ at layer -6 achieved the correlation with human translation quality judgment; marginally better than that using BERT$_{large}$ and XLM-RoBERTa$_{large}$. Therefore, in WMT 2020 metrics shared task *-English MT output evaluation, we submit YiSi-1 scores based on embeddings extracted from the layer -6 of RoBERTa$_{large}$.

3.2 Monolingual models vs. multilingual models

In Figure 1 and 2, we identify a common pattern that for evaluating English, Finnish, French and Chinese, using monolingual models (RoBERTa for English, CamemBERT for French and BERT for Finnish and Chinese) in YiSi-1 achieved the best correlation with human translation quality judgment. The only exception is using German BERT in YiSi-1 for evaluating German MT output where YiSi-1 using XLM-RoBERTa significantly outperforms that using German BERT. One of the possible reasons is that there is a domain mismatch between the training data of the German BERT and the MT output in the evaluation. The data used in the German BERT included 20% of legal documents while the MT output of WMT 2019 metrics shared evaluation set belongs to the news domain. Since YiSi-1 using monolingual BERT model usually outperforms that using multilingual pretrained

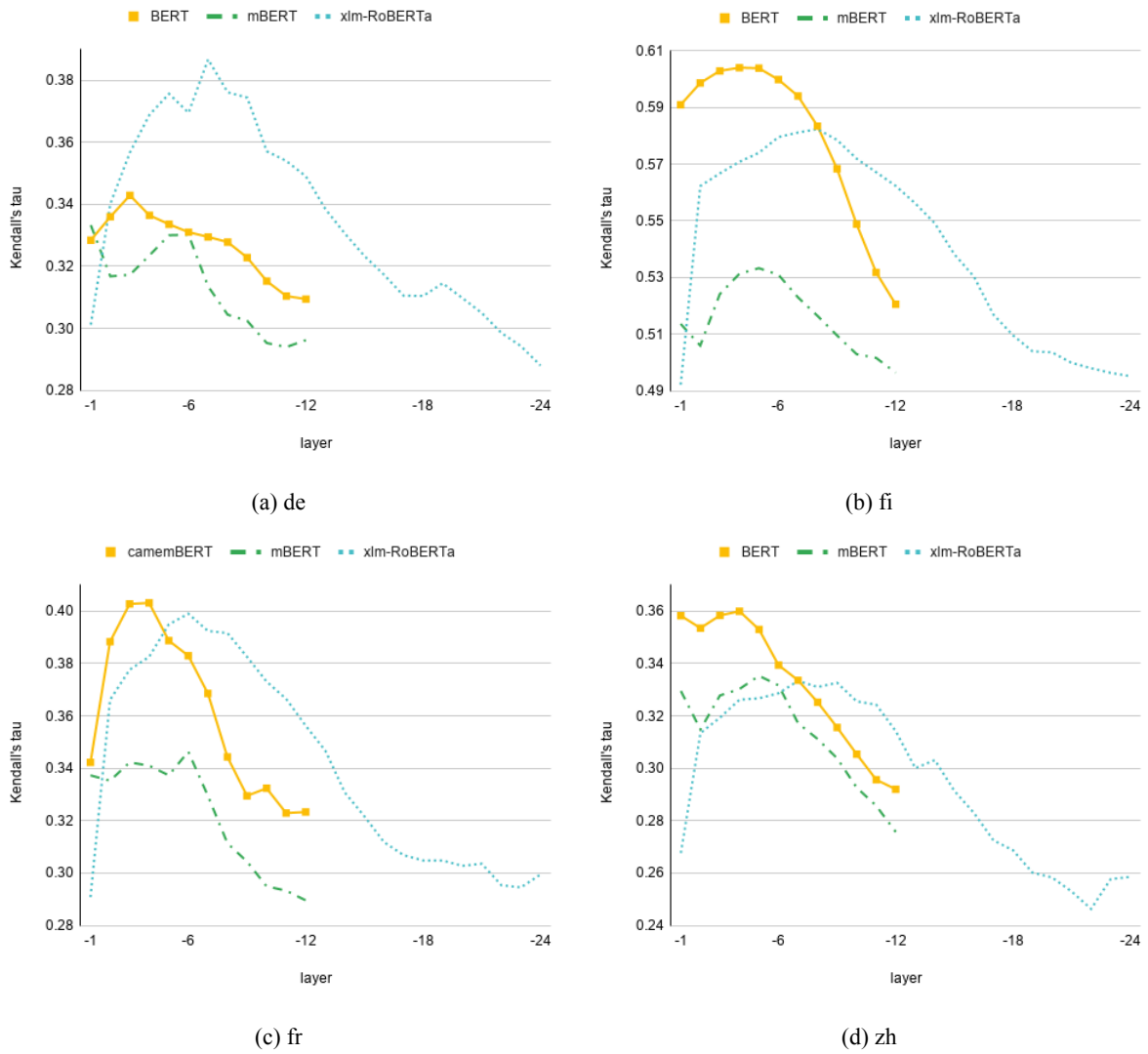


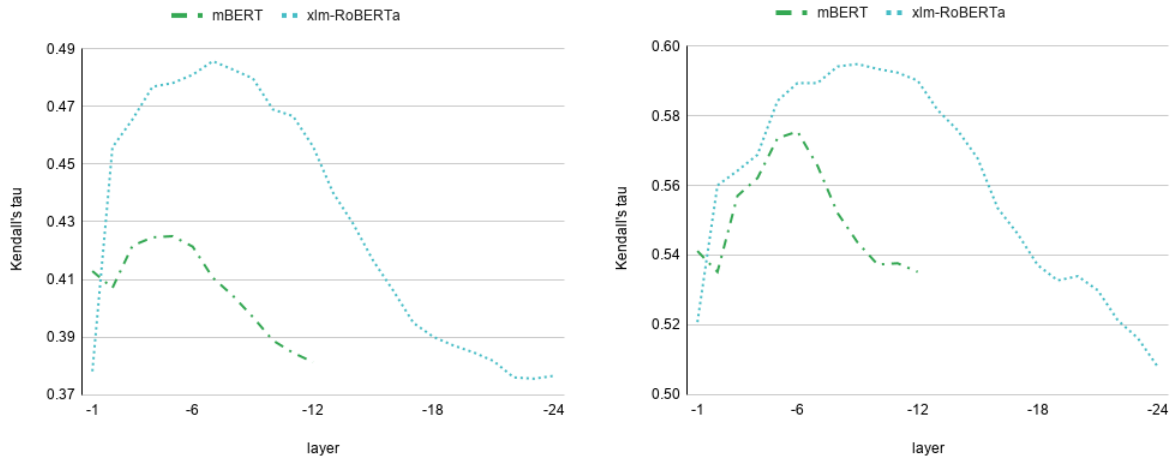
Figure 2: Average segment-level Kendall's τ correlation with human direct assessment on WMT19 (a) *-de, (b) en-fi, (c) de-fr and (d)en-zh news translation test set of YiSi-1 using different pretrained language representation models. Solid lines represent the use of pretrained monolingual models. Dotted line represents the use of pretrained XLM-R and dashed line represents the use of pretrained multilingual BERT.

language model, we believe that for evaluating MT output in WMT 2020 metrics shared task, using YiSi-1 with the monolingual BERT (while available, i.e. CamemBERT for French, BERT for Japanese and Chinese) would be a better model choice.

Another common pattern we see is that YiSi-1 using the monolingual BERT$_{base}$ model usually achieved the best correlation with human translation quality judgment at layer -4. Therefore, in WMT 2020 metrics shared task *-Chinese/French/Japanese MT output evaluation, we submit YiSi-1 scores based on embeddings extracted from the layer -4 of the corresponding monolingual BERT model.

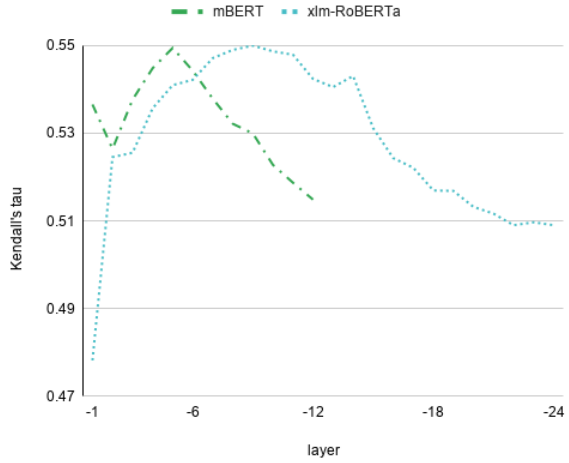
3.3 Multilingual BERT vs. XLM-RoBERTa

In Figure 3, we plot the change of segment-level Kendall's τ correlation for YiSi-1 across different layers of XLM-R and multilingual BERT models for evaluating English-Czech/Gujarati/Kazakh/Lithuanian/Russian.
We identify a common trend, YiSi-1 using embeddings extracted from XLM-RoBERTa significantly outperforms YiSi-1 using embeddings extracted from multilingual BERT, except for evaluating Kazakh MT output where the gains of using XLM-RoBERTa is marginal. On average in all translation directions, the optimal layer of representation in XLM-R for YiSi-1 is layer -7.

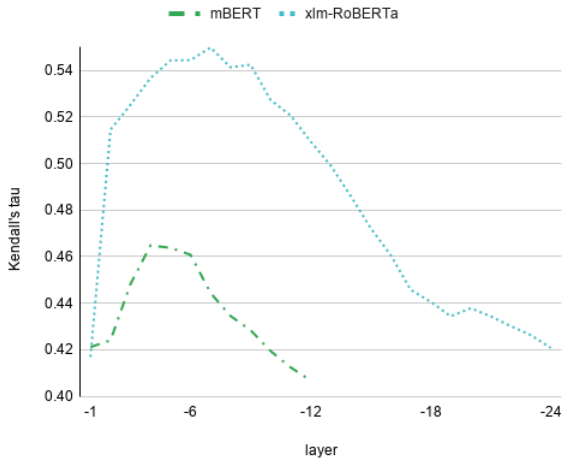


(a) cs

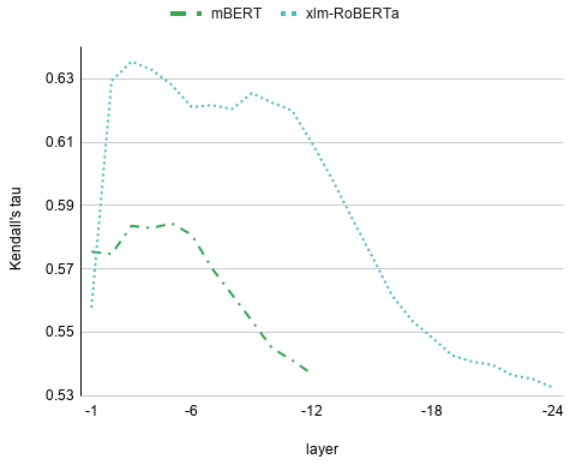
(b) gu



(c) kk



(d) lt



(e) ru

Figure 3: Average segment-level Kendall's τ correlation with human direct assessment on WMT19 (a) *-cs, (b) en-gu, (c) en-kk, (d) en-lt and (e) en-ru news translation test set of YiSi-1 using different pretrained language representation models. Dotted line represents the use of pretrained XLM-R and dashed line represents the use of pretrained multilingual BERT.

Table 1: Kendall's τ correlation of metric scores with the WMT 2019 official human direct assessment judgments at segment level.

input	de	fi	gu	kk	lt	ru	zh	en	en	en	en	en	en	en	en
output	en	en	en	en	en	en	en	cs	de	fi	gu	kk	lt	ru	zh
YiSi-1 (2020)	**.172**	**.354**	**.328**	.425	**.385**	**.230**	**.438**	**.544**	**.384**	**.604**	**.589**	**.547**	**.550**	**.622**	**.360**
YiSi-1 (2019)	.164	.347	.312	**.440**	.376	.217	.426	.475	.351	.537	.551	.546	.470	.585	.355
YiSi-0	.117	.271	.263	.402	.289	.178	.355	.406	.304	.483	.539	.494	.402	.535	.266

Table 2: Kendall's τ correlation of metric scores with the WMT 2019 official human direct assessment judgments at segment level.

input	de	de	fr
output	cs	fr	de
YiSi-1 (2020)	**.427**	**.403**	**.389**
YiSi-1 (2019)	.376	.349	.310
YiSi-0	.331	.296	.277

4 Improvements over previous version of YiSi-1

Table 1 and 2 show the Kendall' s τ correlation with the segment-level human direct assessment relative ranking on the WMT 2019 evaluation set. YiSi-1 (2020) shows consistent and significant improvements when comparing to the previous version of YiSi-1 across all translation directions.

Table 3 and 4 show the Person's ρ correlation with the system-level human direct assessment relative ranking on the WMT 2019 evaluation set. Although the improvements at system-level correlation is less consistent across different translation directions, YiSi-1 (2020) outperforms YiSi-1(2019) in the evaluation of two-third of all the tested translation directions.

5 Conclusion

We have presented an extend study of the pretrained language models used in YiSi-1 for machine translation evaluation. From this study, we conclude that for the best performance of YiSi-1: 1) when evaluating MT output in English, it is recommended to use the contextual embeddings extracted from layer -6 of RoBERTa$_{large}$; 2) when evaluating MT output in languages where monolingual pretrained model in the same or general domain is available, it is recommended to use the contextual embeddings extracted from those models; and finally 3) when evaluating MT output in languages only covered by multilingual pretrained language models, it is recommended to use the contextual embeddings extracted from layer -7 of XLM-RoBERTa.

This improved version of YiSi-1 is submitted to the WMT 2020 metrics shared task. For evaluating Inuktitut↔English where one of the language (Inuktitut) is not covered by any released pretrained langauge model, we build our own XLM cross-lingual language model with the parallel training data.

References

Branden Chan, Timo Möller, Malte Pietsch, Tanay Soni, and Chin Man Yeung. 2019. Open sourcing german bert.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Tohoku University Inui Laboratory. 2019. Pretrained Japanese BERT models. https://github.com/cl-tohoku/bert-japanese.

Eric Joanis, Rebecca Knowles, Roland Kuhn, Samuel Larkin, Patrick Littell, Chi-kiu Lo, Darlene Stewart, and Jeffrey Micher. 2020. The Nunavut Hansard Inuktitut–English parallel corpus 3.0 with preliminary machine translation results. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 2562–2572, Marseille, France. European Language Resources Association.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *Advances in Neural Information Processing Systems (NeurIPS)*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*.

Table 3: Pearson's ρ correlation of metric scores with the WMT 2019 official human direct assessment judgments at system level.

input	de	fi	gu	kk	lt	ru	zh	en	en	en	en	en	en	en	en
output	en	en	en	en	en	en	en	cs	de	fi	gu	kk	lt	ru	zh
YiSi-1 (2020)	**.953**	.987	**.998**	**.991**	.967	.929	**.986**	.971	**.993**	.979	**.945**	**.991**	**.979**	.980	.942
YiSi-1 (2019)	.949	.989	.924	.944	**.981**	**.979**	.979	.962	.991	.971	.909	.985	.963	**.992**	**.951**
YiSi-0	.902	**.993**	.993	**.991**	.927	.958	.937	**.992**	.985	**.987**	.863	.974	.974	.953	.861

Table 4: Pearson's ρ correlation of metric scores with the WMT 2019 official human direct assessment judgments at system level.

input	de	de	fr
output	cs	fr	de
YiSi-1 (2020)	**.981**	.953	**.924**
YiSi-1 (2019)	.973	**.969**	.908
YiSi-0	.978	.952	.820

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Chi-kiu Lo. 2019. YiSi - a unified semantic MT quality evaluation and estimation metric for languages with different levels of available resources. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 507–513, Florence, Italy. Association for Computational Linguistics.

Chi-kiu Lo and Samuel Larkin. 2020. Machine translation reference-less evaluation using yisi-2with bilingual mappings of massive multilingual language model. In *Proceedings of the Fifth Conference on Machine Translation: Shared Task Papers*.

Qingsong Ma, Ondřej Bojar, and Yvette Graham. 2018. Results of the WMT18 metrics shared task: Both characters and embeddings achieve good performance. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 671–688, Belgium, Brussels. Association for Computational Linguistics.

Qingsong Ma, Johnny Wei, Ondřej Bojar, and Yvette Graham. 2019. Results of the wmt19 metrics shared task: Segment-level and strong mt systems pose big challenges. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 62–90, Florence, Italy. Association for Computational Linguistics.

Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric de la Clergerie, Djamé Seddah, and Benoît Sagot. 2020. CamemBERT: a tasty French language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7203–7219, Online. Association for Computational Linguistics.

Nitika Mathur, Timothy Baldwin, and Trevor Cohn. 2019. Putting evaluation in context: Contextual embeddings improve machine translation evaluation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2799–2808, Florence, Italy. Association for Computational Linguistics.

Nitika Mathur, Timothy Baldwin, and Trevor Cohn. 2020. Tangled up in BLEU: Reevaluating the evaluation of automatic machine translation evaluation metrics. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4984–4997, Online. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *40th Annual Meeting of the Association for Computational Linguistics (ACL-02)*, pages 311–318, Philadelphia, Pennsylvania.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Antti Virtanen, Jenna Kanerva, Rami Ilo, Jouni Luoma, Juhani Luotolahti, Tapio Salakoski, Filip Ginter, and Sampo Pyysalo. 2019. Multilingual is not enough: Bert for finnish. *arXiv preprint arXiv:1912.07076*.

Wietse de Vries, Andreas van Cranenburgh, Arianna Bisazza, Tommaso Caselli, Gertjan van Noord, and Malvina Nissim. 2019. BERTje: A Dutch BERT model.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5753–5763.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Machine Translation Reference-less Evaluation using YiSi-2 with Bilingual Mappings of Massive Multilingual Language Model

Chi-kiu Lo and **Samuel Larkin**
Multilingual Text Processing
Digital Technologies Research Centre
National Research Council Canada (NRC-CNRC)
1200 Montreal Road, Ottawa, ON K1A 0R6, Canada
{chikiu.lo,samuel.larkin}@nrc-cnrc.gc.ca

Abstract

We present a study on using YiSi-2 with massive multilingual pretrained language models for machine translation (MT) reference-less evaluation. Aiming at finding better semantic representation for semantic MT evaluation, we first test YiSi-2 with contextual embeddings extracted from different layers of two different pretrained models, multilingual BERT and XLM-RoBERTa. We also experiment with learning bilingual mappings that transform the vector subspace of the source language to be closer to that of the target language in the pretrained model to obtain more accurate cross-lingual semantic similarity representations. Our results show that YiSi-2's correlation with human direct assessment on translation quality is greatly improved by replacing multilingual BERT with XLM-RoBERTa and projecting the source embeddings into the target embedding space using a cross-lingual linear projection (CLP) matrix learnt from a small development set.

1 Introduction

The machine translation quality estimation as a metric (QE as a metric) task was first introduced in WMT 2019 (Ma et al., 2019; Fonseca et al., 2019) to encourage the exploration of reference-less evaluation metrics. QE as a metric task shifts the use case of the QE systems from assisting professional translators to estimate post-editing efforts to assisting MT developers or general MT users to discriminate the translation quality of different MT systems without the presence of a human reference translation. YiSi-2, the reference-less variants of the YiSi metric (Lo, 2019), was the only metric who participated in evaluating all the translation directions in WMT 2019 QE as a metric shared task.

The QE as a metric task is very similar to Task 1 (Sentence-level direct assessment) of WMT20's quality estimation shared task where metric performance is evaluated in terms of correlation at the sentence-level with human direct assessment scores on translation quality. The subtle but crucial difference between the WMT20 QE Task 1 and the QE as a metric task is that QE systems for the former task is trained specifically to estimate the quality of a single MT system whereas QE metrics for the latter task is generalized for multiple machine translation systems. The QE systems for WMT20's QE Task 1 have access to the MT system that generate the translations while the reference-less metrics for the latter task have no information on the MT systems being evaluated.

In WMT 2019 metrics shared task, pretrained multilingual BERT (Devlin et al., 2018) was used in YiSi for both MT reference-based (YiSi-1) and reference-less (YiSi-2) evaluation in all tested translation directions where monolingual pretrained BERT model was not available for the target language (such as Czech, German, etc.). Since then, another massive multilingual pretrained language model, XLM-RoBERTa (Conneau et al., 2020), has been published. We evaluate the use of contextual embeddings extracted from each of the intermediate layers of the two models in MT reference-less evaluation.

In addition, despite using the same pretrained embedding model of last year, YiSi-2 showed a significant performance degradation when comparing to YiSi-1. For example, segment-level correlation with human direct assessment for evaluating English→Czech drops from 0.475 (YiSi-1) to 0.069 (YiSi-2). This shows that the cross-lingual semantic representation in pretrained multilingual BERT is not as accurate as the monolingual semantic representation for each language. In other words, we observed the language clustering effect where a clear segregation of vector subspace among different languages in the multilingual contextual em-

Proceedings of the 5th Conference on Machine Translation (WMT), pages 903–910
Online, November 19–20, 2020.

bedding model. Inspired by Zhao et al. (2020), we employ a weakly-supervised bilingual mapping learnt from a small development set that transforms the contextual embeddings of the source sentence to the target subspace for better cross-lingual semantic similarity evaluation.

In this paper, we show that YiSi-2's correlation with human direct assessment on translation quality is greatly improved by replacing multilingual BERT with XLM-RoBERTa$_{large}$ using the optimal intermediate layer (7^{th} layer count from the last) and projecting the source embeddings into the target embedding space using a cross-lingual linear projection matrix learnt from a small development set.

2 YiSi-2

YiSi (Lo, 2019) is a unified semantic MT quality evaluation and estimation metric for languages with different levels of available resources. YiSi-1 measures the similarity between a machine translation and human references by aggregating weighted distributional (lexical) semantic similarities, and optionally incorporating shallow semantic structures. Improvements in YiSi-1 for WMT 2020 metrics shared task is detailed in (Lo, 2020).

YiSi-2 is the bilingual, reference-less version, which uses bilingual word embeddings to evaluate cross-lingual lexical semantic similarity between the input and MT output.

2.1 Massive Multilingual Pretrained Language Models

YiSi-2 relies on a cross-lingual language representation to evaluate the cross-lingual lexical semantic similarity. Previously, it used pretrained multilingual BERT (Devlin et al., 2018) for this purpose. BERT captures the sentence context in the embeddings, such that the embedding of the same subword unit in different sentences would be different from each other and be better represented in the embedding space. Since multilingual BERT is trained on the concatenation of non-parallel data from each language, the circular dependency deadlock between parallel resource and cross-lingual semantic similarity is broken (Lo and Simard, 2019). Multilingual BERT covers the 104 largest languages in Wikipedia.

XLM-RoBERTa (Conneau et al., 2020) (XLM-R) is also a massive multilingual pretrained language model. Similar to BERT, XLM-R is also trained with a masked language model task on the concatenation of non-parallel data. The differences between XLM-R and BERT are 1) XLM-R is trained on the CommonCrawl corpus which is significantly larger than the Wikipedia training data used by BERT; 2) instead of a uniform data sampling rate used in BERT, XLM-R uses a language sampling rate that is proportional to the amount of data available in the training set. Because of these differences, XLM-R performs better on low resource languages than multilingual BERT. XLM-R covers 100 languages. In this work, we use XLM-R$_{large}$ for the best performance on cross-lingual semantic similarity.

As suggested by Devlin et al. (2018); Peters et al. (2018); Zhang et al. (2020), we experimented using contextual embeddings extracted from different layers of the multilingual language encoder to find out the layer that best represents the semantic space of the language.

2.2 Inuktitut-English Cross-lingual Language Model

Since Inuktitut is neither covered by pretrained multilingual BERT nor XLM-RoBERTa, we trained our own Inuktitut-English XLM (Lample and Conneau, 2019) using the Nunavut Hansard 3.0 (NH) parallel corpus (Joanis et al., 2020). The model was trained with masked language model and translation language model tasks. The Inuktitut-English XLM model has 12 layers with 8 heads and embedding size of 512.

2.3 Cross-lingual Linear Projection

In the WMT 2019 metrics shared task (Ma et al., 2019), we saw a very significant performance degradation between YiSi-1 and YiSi-2. This shows that current multilingual language models construct a shared multilingual space in an unsupervised manner without any direct bilingual signal, in which representations of context in the same language are likely to cluster together in part of the subspace and there is a language segregation in the shared multilingual space. Inspired by Artetxe et al. (2016) and Zhao et al. (2020), we obtain subword token pairs from the news translation task development set for each language (each contains around 1k to 3k sentence pairs) aligned by maximum alignment of their semantic similarities. We then train a cross-lingual linear projection (Zhao et al., 2020) that transforms the source embeddings into the target embeddings subspace.

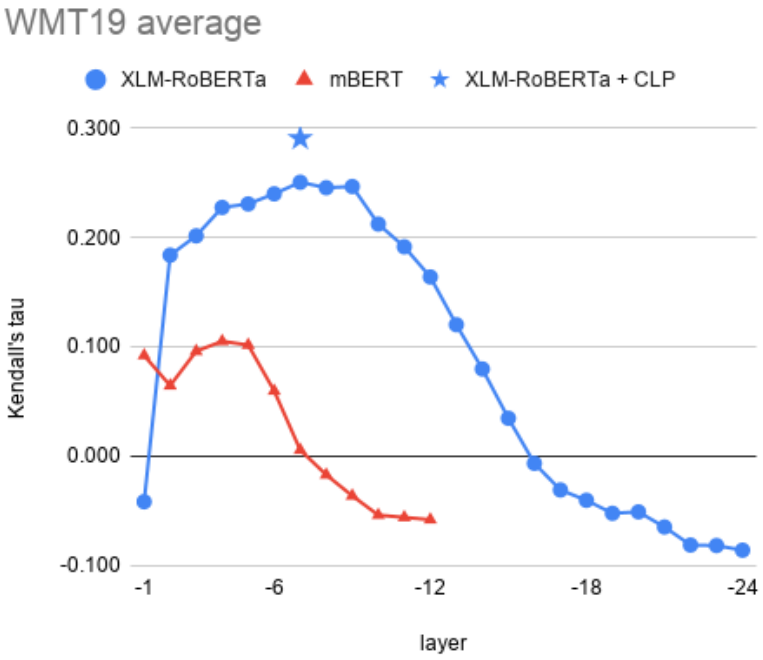


Figure 1: Segment-level Kendall's τ correlation with human direct assessment averaged over all WMT 2019 news translation test sets of YiSi-2 using contextual embeddings extracted from different layers of the multilingual pre-trained language models. On the x-axis, layer $-n$ means, YiSi-2 based on the embeddings of the n^{th} layer, counting from the last, of XLM-RoBERTa$_{large}$ (blue circles), multilingual BERT (red triangles) and layer -7 of of XLM-RoBERTa$_{large}$ with source embeddings projected to target language space using CLP (blue star).

Table 1: Segment-level Kendall's τ correlation of metric scores with the WMT 2019 official human direct assessment judgments.

input	de	fi	gu	kk	lt	ru	zh	en	en	en	en	en	en	en	en
output	en	en	en	en	en	en	en	cs	de	fi	gu	kk	lt	ru	zh
Reference-based evaluation metric															
YiSi-1 (2019)	.164	.347	.312	.440	.376	.217	.426	.475	.351	.537	.551	.546	.470	.585	.355
YiSi-0	.117	.271	.263	.402	.289	.178	.355	.406	.304	.483	.539	.494	.402	.535	.266
sentBLEU	.056	.233	.188	.377	.262	.125	.323	.367	.248	.396	.465	.392	.334	.469	.270
QE as a metric															
YiSi-2 (2020)	.116	.271	.249	.370	.281	.121	.340	.299	.329	.459	.512	.459	.314	.078	.158
YiSi-2 (2019)	.068	.126	-.001	.096	.075	.053	.253	.069	.212	.239	.147	.187	.003	-.155	.044

Table 2: Segment-level Kendall's τ correlation of metric scores with the WMT 2019 official human direct assessment judgments.

input	de	de	fr
output	cs	fr	de
Reference-based evaluation metric			
YiSi-1 (2019)	.376	.349	.310
YiSi-0	.331	.296	.277
sentBLEU	.203	.235	.179
QE as a metric			
YiSi-2 (2020)	.355	.294	.226
YiSi-2 (2019)	.199	.186	.066

3 Results

We use WMT 2019 metrics task evaluation set (Ma et al., 2019) for our development experiments. The official human judgments for translation quality of WMT 2019 were collected using reference-based direct assessment.

Since we use exactly the same correlation analysis as the official metrics shared evaluation and the 2019 version of YiSi performed consistently well among participants in WMT 2019, we only compare our results with the 2019 version of YiSi and BLEU. Our results are directly comparable with those reported in Ma et al. (2019).

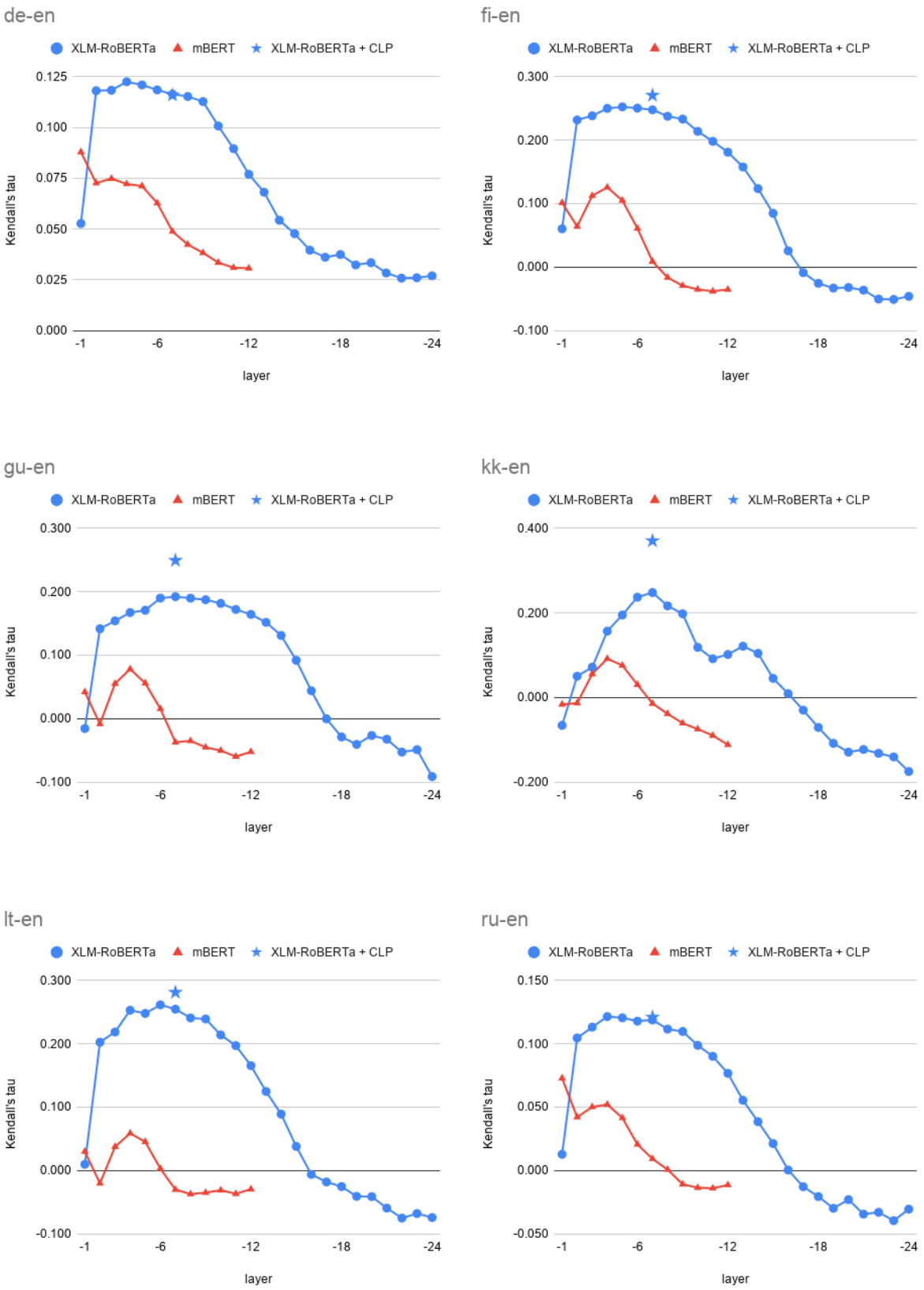


Figure 2: Segment-level Kendall's τ correlation with human direct assessment on WMT 2019 de-en, fi-en, gu-en, kk-en, lt-en and ru-en news translation test set of YiSi-2 using contextual embeddings extracted from different layers of the multilingual pretrained language models. On the x-axis, layer $-n$ means YiSi-2 based on the embeddings of the n^{th} layer, counting from the last, of XLM-RoBERTa$_{large}$ (blue circles), multilingual BERT (red triangles) and layer -7 of of XLM-RoBERTa$_{large}$ with source embeddings projected to target language space using CLP (blue star).

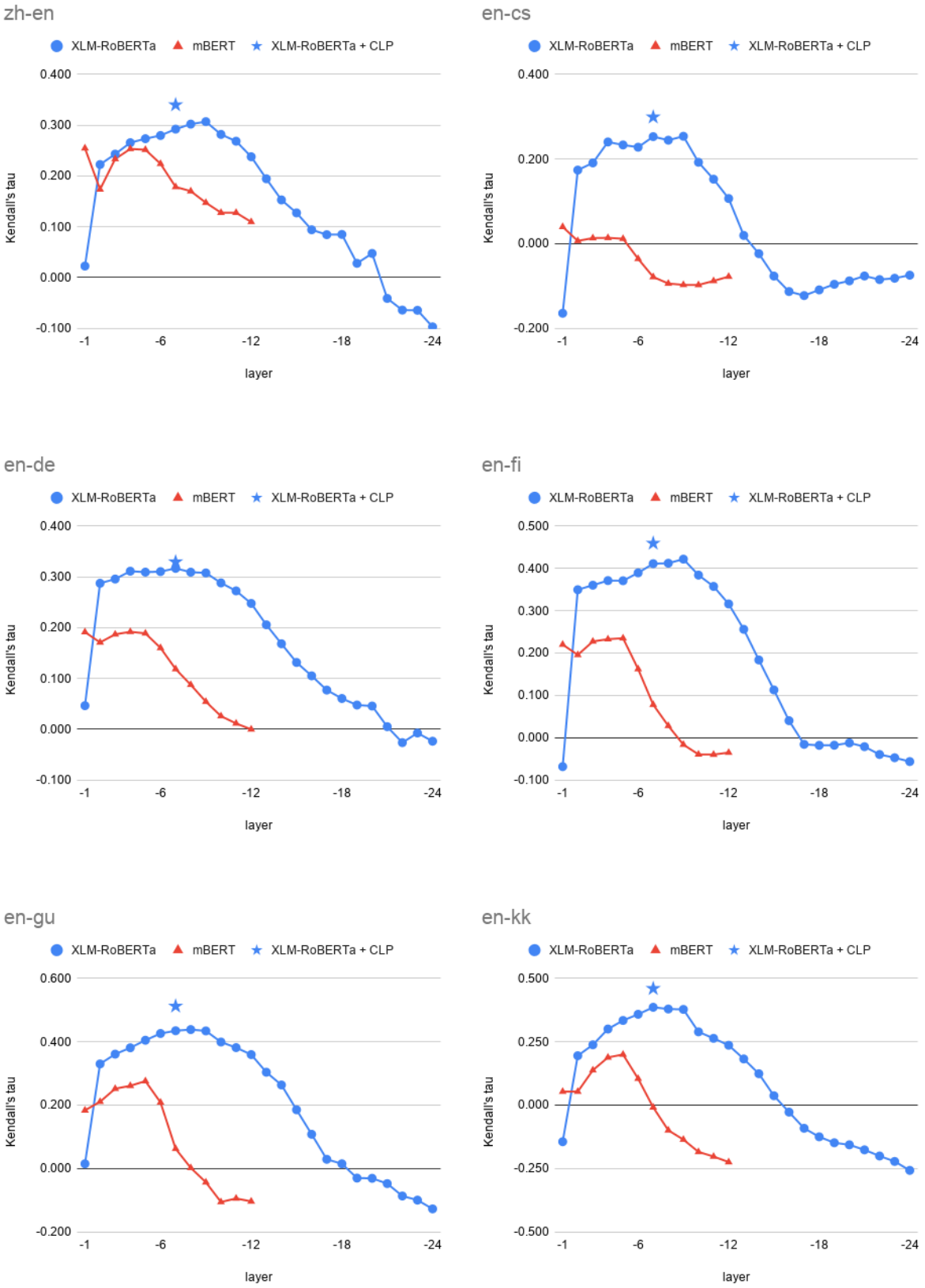


Figure 3: Segment-level Kendall's τ correlation with human direct assessment on WMT 2019 zh-en, en-cs, en-de, en-fi, en-gu and en-kk news translation test set of YiSi-2 using contextual embeddings extracted from different layers of the multilingual pretrained language models. On the x-axis, layer $-n$ means YiSi-2 based on the embeddings of the n^{th} layer, counting from the last, of XLM-RoBERTa$_{large}$ (blue circles), multilingual BERT (red triangles) and layer -7 of of XLM-RoBERTa$_{large}$ with source embeddings projected to target language space using CLP (blue star).

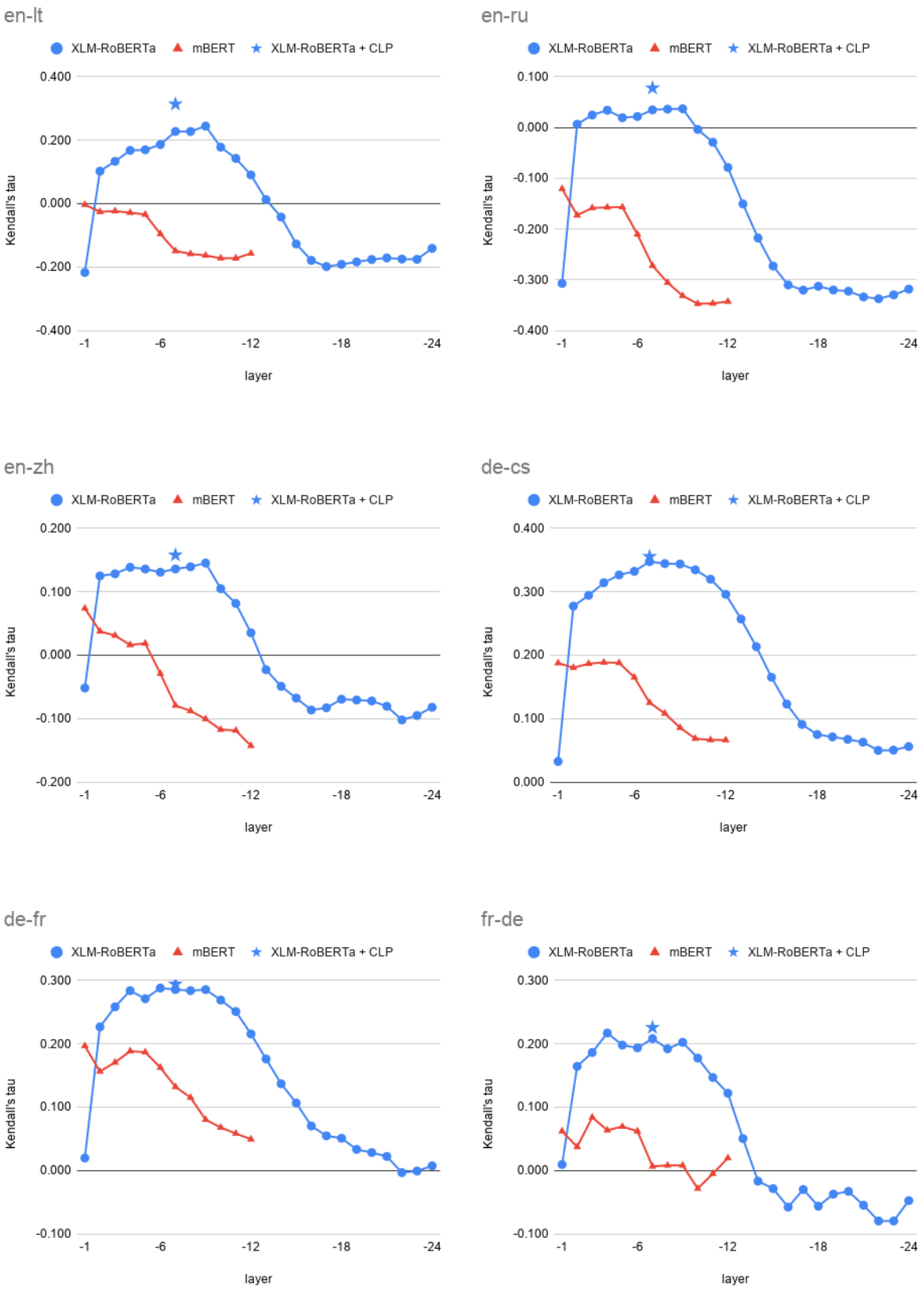


Figure 4: Segment-level Kendall's τ correlation with human direct assessment on WMT 2019 en-lt, en-ru, en-zh, de-cs, de-fr and fr-de news translation test set of YiSi-2 using contextual embeddings extracted from different layers of the multilingual pretrained language models. On the x-axis, layer $-n$ means YiSi-2 based on the embeddings of the n^{th} layer, counting from the last, of XLM-RoBERTa$_{large}$ (blue circles), multilingual BERT (red triangles) and layer -7 of of XLM-RoBERTa$_{large}$ with source embeddings projected to target language space using CLP (blue star).

Table 3: System-level Pearson's ρ correlation of metric scores with the WMT 2019 official human direct assessment judgments.

input	de	fi	gu	kk	lt	ru	zh	en	en	en	en	en	en	en	en
output	en	en	en	en	en	en	en	cs	de	fi	gu	kk	lt	ru	zh
Reference-based evaluation metric															
YiSi-1 (2019)	.949	.989	.924	.944	.981	.979	.979	.962	.991	.971	.909	.985	.963	.992	.951
YiSi-0	.902	.993	.993	.991	.927	.958	.937	.992	.985	.987	.863	.974	.974	.953	.861
BLEU	.849	.982	.834	.946	.961	.879	.899	.897	.921	.969	.737	.852	.989	.986	.901
QE as a metric															
YiSi-2 (2020)	.898	.959	.739	.981	.935	.461	.980	.773	.963	.906	.890	.977	.761	.473	.449
YiSi-2 (2019)	.796	.642	.566	.324	.442	.339	.940	.324	.924	.696	.314	.339	.055	.766	.097

Table 4: System-level Pearson's ρ correlation of metric scores with the WMT 2019 official human direct assessment judgments.

input	de	de	fr
output	cs	fr	de
Reference-based evaluation metric			
YiSi-1 (2019)	.973	.969	.908
YiSi-0	.978	.952	.820
BLEU	.941	.891	.864
QE as a metric			
YiSi-2 (2020)	.860	.853	.461
YiSi-2 (2019)	.606	.721	.530

3.1 Segment-level correlation with human judgment

In Figure 1, 2, 3 and 4, we plot the change of segment-level Kendall's τ correlation for YiSi-2 across different layers of XLM-R and multilingual BERT models. We identify a common trend, YiSi-2 using embeddings extracted from XLM-R significantly outperforms YiSi-2 using embeddings extracted from multilingual BERT. From figure 1, we see that, on average, on all translation directions, the optimal layer of representation in XLM-R for YiSi-2 is layer -7. Learning the cross-lingual linear projection matrix to transform the source embeddings into the target language subspace shows a greater improvement overall. This is our "YiSi-2 (2020)" submission to the QE as a metric task.

Table 1 and 2 show the Kendall's τ correlation with the segment-level human direct assessment relative ranking on the WMT 2019 evaluation set. YiSi-2 (2020) shows consistent and significant improvements when comparing to the previous version of YiSi-2 across all translation directions.

Although YiSi-2 (2020) still performs worse than YiSi-1, YiSi-2 (2020) correlates better with human judgment than the reference-based metric, sentBLEU, and its performances are comparable to those of the character-based YiSi variant, YiSi-0, on evaluating translation quality for most of the translation directions.

3.2 Correlation with human judgment at system level

Table 3 and 4 show the Person's ρ correlation with the system-level human direct assessment relative ranking on the WMT 2019 evaluation set.

Similar to the segment-level results, although YiSi-2 (2020) still performs significantly worse than YiSi-1, we observe significant improvements, compared to the previous version of YiSi-2, consistently across all translation directions. We also show that by replacing the multilingual BERT with XLM-R and using bilingual mappings to better align the source and target language subspaces in XLM-R, YiSi-2 (2020) correlates better with human judgment than the reference-based metric, BLEU, on evaluating translation quality for most of the translation directions.

4 Conclusion

We have presented an improved version of YiSi-2 that uses XLM-RoBERTa and a cross-lingual linear projection of the source embedding to the target language subspace to better capture the semantic representation across languages. Our results show that YiSi-2 correlates better with human judgement on evaluating translation quality than BLEU for most of the evaluation conditions. This improved version of YiSi-2 is submitted to the WMT 2020 Metrics shared task QE as a metric track. For evaluating Inuktitut↔English where one of the language (Inuktitut) is not covered by XLM-R, we build our own XLM cross-lingual language model with the parallel training data. Potential research directions definitely include improving massive multilingual pretrained language model to close the performance gap between YiSi-1 and YiSi-2 and expanding the language coverage of these models in post-hoc and unsupervised manner.

References

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2016. Learning principled bilingual mappings of word embeddings while preserving monolingual invariance. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2289–2294, Austin, Texas. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Erick Fonseca, Lisa Yankovskaya, André F. T. Martins, Mark Fishel, and Christian Federmann. 2019. Findings of the WMT 2019 shared tasks on quality estimation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 1–10, Florence, Italy. Association for Computational Linguistics.

Eric Joanis, Rebecca Knowles, Roland Kuhn, Samuel Larkin, Patrick Littell, Chi-kiu Lo, Darlene Stewart, and Jeffrey Micher. 2020. The Nunavut Hansard Inuktitut–English parallel corpus 3.0 with preliminary machine translation results. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 2562–2572, Marseille, France. European Language Resources Association.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *Advances in Neural Information Processing Systems (NeurIPS)*.

Chi-kiu Lo. 2019. YiSi - a unified semantic MT quality evaluation and estimation metric for languages with different levels of available resources. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 507–513, Florence, Italy. Association for Computational Linguistics.

Chi-kiu Lo. 2020. Extended study on using pretrained language models and YiSi-1 for machine translation evaluation. In *Proceedings of the Fifth Conference on Machine Translation: Shared Task Papers*.

Chi-kiu Lo and Michel Simard. 2019. Fully unsupervised crosslingual semantic textual similarity metric based on BERT for identifying parallel data. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 206–215, Hong Kong, China. Association for Computational Linguistics.

Qingsong Ma, Johnny Wei, Ondřej Bojar, and Yvette Graham. 2019. Results of the wmt19 metrics shared task: Segment-level and strong mt systems pose big challenges. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 62–90, Florence, Italy. Association for Computational Linguistics.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Wei Zhao, Goran Glavaš, Maxime Peyrard, Yang Gao, Robert West, and Steffen Eger. 2020. On the limitations of cross-lingual encoders as exposed by reference-free machine translation evaluation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1656–1671, Online. Association for Computational Linguistics.

Unbabel's Participation in the WMT20 Metrics Shared Task

Ricardo Rei **Craig Stewart** **Ana C Farinha** **Alon Lavie**
Unbabel AI
{ricardo.rei, craig.stewart, catarina.farinha, alon.lavie}@unbabel.com

Abstract

We present the contribution of the Unbabel team to the WMT 2020 Shared Task on Metrics. We intend to participate on the segment-level, document-level and system-level tracks on all language pairs, as well as the "QE as a Metric" track. Accordingly, we illustrate results of our models in these tracks with reference to test sets from the previous year. Our submissions build upon the recently proposed COMET framework: we train several estimator models to regress on different human-generated quality scores and a novel ranking model trained on relative ranks obtained from Direct Assessments. We also propose a simple technique for converting segment-level predictions into a document-level score. Overall, our systems achieve strong results for all language pairs on previous test sets and in many cases set a new state-of-the-art.

1 Introduction

In this paper we describe our submission to the WMT20 Metrics shared task. Our work is based on the COMET[1] framework, as presented in Rei et al. (2020), and extended here to evaluation of MT output at segment, document and system-level, forming the basis of our submissions to the corresponding task tracks. Recently, automatic evaluation of MT has followed most other sub-fields in NLP with a notable interest in leveraging the power of large, pre-trained language models. Metrics such as BERT REGRESSOR (Shimanaka et al., 2019), BERTSCORE (Zhang et al., 2020), BLEURT (Sellam et al., 2020) and our more recent COMET (Rei et al., 2020), all build upon developments in language modelling to generate automatic metrics with high correlation with human judgement. Our MT evaluation models follow a similar strategy, specifically utilizing the most recent iterations of the XLM-RoBERTa model presented in Conneau et al. (2020).

The uniqueness of our approach comes from our inclusion of the source text as input which was demonstrated in Takahashi et al. (2020) and Rei et al. (2020) to be beneficial to the model. In our contribution to the shared task, we demonstrate methods of further exploiting information in the source text as well as a technique to fully harness the power of pre-trained language models to further improve the prediction accuracy of our evaluation framework when more than one reference translation is available.

For the shared task, we utilize two primary types of models built using the COMET framework, namely; the Estimator models, which regress directly on human scores of MT quality such as Direct Assessment; and the COMET-RANK (base) model used to rank MT outputs and systems.

In addition to the models themselves, we also make the following research contributions:

1. We introduce a method for handling multiple references at inference time and for optimizing the utility of information from all available text inputs

2. We propose a simple technique for calculating a document-level score from a weighted average of segment-level scores

We demonstrate that our COMET framework trained models achieve state-of-the-art results or are competitive on all settings introduced in the WMT19 Metrics shared task, outperforming, in some cases, more recently proposed metrics such as BERTSCORE (Zhang et al., 2020), BLEURT (Sellam et al., 2020) and PRISM (Thompson and Post, 2020).

[1] **C**rosslingual **O**ptimized **M**etric for **E**valuation of **T**ranslation hosted at: https://github.com/Unbabel/COMET

Proceedings of the 5th Conference on Machine Translation (WMT), pages 911–920
Online, November 19–20, 2020.

2 The COMET Framework

As outlined in Rei et al. (2020), the COMET framework allows for training of specialized evaluation metrics that correlate well with different types of human-generated quality scores. The general structure of the framework consists of a cross-lingual encoder that produces a series of token-level vector embeddings for source, hypothesis and reference inputs, a pooling layer which converts the various token-level representations into segment-level vectors for each input, and a predictive neural network that generates a quality score. The latter model can either be trained to regress directly on a score to produce predictions of segment-level quality, or can be trained as a ranker to differentiate MT systems. In our contribution to the shared task, we introduce two varieties of models built on the COMET framework that are extensions of the models evaluated in Rei et al. (2020).

3 COMET Models

3.1 Estimator Models

Our Estimators generally follow the architecture proposed in Rei et al. (2020), that is to say we encode segment-level representations using XLM-RoBERTa and pass these outputs through a feed-forward regressor. As in Rei et al. (2020), we train three versions of this basic estimator model against different types of human judgement; *Human-mediated Translation Edit Rate* (HTER) (Snover et al., 2006), a proprietary implementation of *Multidimensional Quality Metric* (MQM) (Lommel et al., 2014) and (in-line with the present task) *Direct Assessments* (DA) (Graham et al., 2013). The hyper-parameters used for these models are exactly as described in Rei et al. (2020), excluding the following alterations: we use XLM-RoBERTa large instead of base and we increase the feed-forward hidden sizes (from 2304 in the first layer and 1152 in the second to 3072 and 1536 hidden units, respectively). We also keep the embedding layer frozen and apply a layer-wise learning rate decay (as proposed in Howard and Ruder (2018)) by which each transformer layer has a learning rate scaled at 0.95 times the rate of the layer above. By doing this, we hope that our metric generalizes better to new language pairs introduced this year.

3.2 Translation Ranking Model

While for the Estimators using a larger pretrained encoder seems to improve performance we found that for the Translation Ranking Model, larger pretrained encoders lead to training instability and an overall worse performance. For that reason we decided to keep the model proposed in (Rei et al., 2020) without any alteration.

4 Corpora

Below we provide an outline of the various datasets used to train our models:

4.1 HTER Corpora

Our HTER corpus is a concatenation of two publicly available corpora: the QT21 corpus and the APE-QUEST corpus. While the QT21 corpus contains segments from the information technology and life sciences domains (Specia et al., 2017), the APE-QUEST contains segments from the legal domain (Ive et al., 2020). Concatenation of these two corpora gives a total of 211K tuples with source sentence, corresponding human-generated reference, MT hypothesis, and post-edited MT (PE). With regard to the language pairs in each corpus, QT21 covers: English to German (en-de), Latvian (en-lt) and Czech (en-cs), and German to English (de-en); while APE-QUEST covers: English-Dutch (en-nl), English-French (en-fr), English-Portuguese (en-pt). Finally, the HTER score is obtained by calculating the translation edit rate (TER) (Snover et al., 2006) between the MT hypothesis and the corresponding PE. By doing this, we were able to create a large HTER corpus covering several language pairs and different domains.

4.2 MQM Corpus

Our MQM corpus is an extension of the proprietary corpus presented in Rei et al. (2020). This internal data consists of customer support chat messages translated using a domain adapted MT model and their corresponding references (consisting of post-edited translations from earlier iterations of the MT systems). The data was then MQM-annotated according to the guidelines set out in Burchardt and Lommel (2014). Our final corpus contains 27K tuples from English into 15 different languages and/or dialects: German (en–de), Spanish (en–es), Latin-American Spanish (en–es-latam), French (en–fr), Italian (en–it), Japanese (en–ja), Dutch (en–nl), Portuguese (en–pt), Brazilian Portuguese (en–pt-br), Russian (en–ru), Swedish (en–sv), Turkish (en–tr), Polish (en–pl), simplified Chinese (en–zh-CN), and Taiwanese Chinese (en–zh-TW).

4.3 DA Corpora

Every year, since 2008, the WMT News Translation shared task organizers collect human judgements in the form of DAs. Since 2017, due to a lack of annotators, these scores are mapped to relative rankings (DARR). We take advantage of this data in two ways: 1) we use the scores directly in order to train an estimator model, 2) as in Rei et al. (2020), we use the DARR to train a translation ranking system. The collective corpora of 2017, 2018 and 2019 contain a total of 24 language pairs, including low-resource languages such as English-Gujarati (en-gu) and English-Kazakh (en-kk). For the purposes of this paper we use the data from 2017 and 2018 to train and the data from 2019 to validate. Later, for participation in the 2020 shared task, we intend to include the data from 2019 in our training corpus.

5 Segment-Level Task

At segment-level, we take each of our Estimator models trained to predict MQM, HTER and DA and predict segment-level scores on the DARR data from WMT19. We then generate pairwise rankings based on these predicted scores. For each language pair we apply the formulation of Kendall's Tau (τ) from the shared task (Ma et al., 2019) as follows:

$$\tau = \frac{\textit{Concordant} - \textit{Discordant}}{\textit{Concordant} + \textit{Discordant}} \quad (1)$$

Concordant here being the number of times a metric assigns a higher score to the "better" hypothesis h^+ and *Discordant*, the number of times a metric assigns a higher score to the "worse" hypothesis h^-, or that the evaluation was otherwise equal.

6 Document-Level Task

In the WMT2019 News Translation the organizers introduced a document-level translation task (Barrault et al., 2019) for en-de and en-cs. This means that for those language pairs we are able to obtain document-level direct assessments. We can compute a score taking into account an entire document and correlate it with the human evaluation also carried out at document-level.

For our document-level submission we propose the generation of a document-level score as a weighted average of the predicted scores for each segment composing that document (hereinafter called micro-average score), where the same is weighted by segment length.

To calculate this score at inference time we pass the entire document (divided into segments) through the model as a single batch. This has the added effect of reducing inference time.

7 System-level Task

Following previous years, the metric used in the System-level Task will be Pearson's r correlation score. The correlation is calculated between the average of all DA human z-scores for a given system and language pair, and the average of the corresponding scores predicted by a given metric. Because the goal of some metrics is to maximize the correlation with human judgements (i.e. BLEU), while for others is to minimize that correlation (i.e. HTER), the value reported is its absolute value.

7.1 Robustness to high-permorming systems

One important finding from WMT19 is the general deterioration of metrics' performance when considering only the top n MT systems (Ma et al., 2019). Previously, we showed robustness of our metrics in this scenario in terms of Kendall's Tau at segment-level (Rei et al., 2020). Mathur et al. (2020) show that at system-level, Pearson correlation is highly influenced by outliers and that performances for metrics such as BLEU drop significantly when considering only the top systems. To address this, we propose a system-level pairwise comparison measured with the same Kendall's Tau formulation used for segment-level analysis outlined in section 5 above. By doing this, we are not only better handling possible outliers, but emulating a real world application of these metrics: In most cases (both in academia and industry), we want a metric that can successfully differentiate between two systems, even if those systems are very close in terms of performance, which is often the case.

8 Quality Estimation as a Metric

Given the clear parallels between the COMET framework and modern approaches to Quality Estimation such as Kepler et al. (2019), we used our framework to participate in the "QE as a Metric" track of the shared task by removing the reference at input and proportionately reducing the dimensions of the feed-forward network to accommodate the reduced input.

Table 1: Segment-level Kendall's Tau (τ) correlations for language pairs from English-to-other for the WMT19 Metrics DARR corpus.

	en-cs	**en-de**	**en-fi**	**en-gu**	**en-kk**	**en-lt**	**en-ru**	**en-zh**	
Nº Tuples	27178	99840	31820	11355	18172	17401	24334	18658	**avg.**
BLEU	0.364	0.248	0.395	0.463	0.363	0.333	0.4691	0.235	0.410
CHRF	0.444	0.321	0.518	0.548	0.510	0.438	0.548	0.241	0.510
BERTSCORE (F1)	0.486	0.350	0.526	0.559	0.534	0.464	0.581	0.350	0.550
PRISM	0.580	0.416	0.590	-	0.529	0.555	0.581	0.373	0.518
COMET-MQM (large)	0.595	0.405	0.594	0.580	0.546	0.607	0.693	0.400	0.553
COMET-HTER (large)	0.610	0.427	0.610	0.587	0.569	0.615	0.707	0.405	0.566
COMET-DA (large)	**0.618**	**0.435**	0.620	**0.617**	0.585	0.619	**0.711**	0.427	0.579
COMET-RANK (base)	0.603	0.427	**0.664**	0.611	**0.693**	**0.665**	0.580	**0.449**	**0.587**

Table 2: Segment-level Kendall's Tau (τ) correlations on language pairs with English as a target for the WMT19 Metrics DARR corpus.

	de-en	**fi-en**	**gu-en**	**kk-en**	**lt-en**	**ru-en**	**zh-en**	
Nº Tuples	85365	32179	20110	9728	21862	39852	31070	**avg.**
BLEU	0.054	0.236	0.194	0.276	0.249	0.115	0.321	0.206
CHRF	0.123	0.292	0.240	0.323	0.304	0.177	0.371	0.261
BERTSCORE (F1)	0.191	0.354	0.292	0.351	0.381	0.221	0.433	0.318
BLEURT (large-512)	0.174	**0.374**	0.313	0.372	0.388	0.220	0.436	0.325
PRISM	0.189	0.366	**0.320**	0.362	0.382	0.220	0.434	0.325
COMET-MQM (large)	0.191	0.360	0.289	0.346	0.373	0.213	0.426	0.314
COMET-HTER (large)	0.193	0.351	0.286	0.340	0.375	0.209	0.429	0.312
COMET-DA (large)	**0.220**	0.368	0.316	**0.378**	**0.405**	**0.231**	**0.462**	**0.340**
COMET-RANK (base)	0.202	**0.399**	**0.341**	0.358	**0.407**	0.180	0.445	0.333

9 Multi-Reference Handling

In this year's shared task we are provided with a second human-generated reference for German-to-English (de-en), Russian-English (ru-en) and Chinese-to-English (zh-en). Given that our base framework currently supports the input of only one single reference, we introduce a method of leveraging information from a second reference at inference time.

During standard training of our models, we input source, hypothesis and reference in that order, resulting in a concatenation of embeddings as detailed further in Rei et al. (2020). During training, with a probability of $p = 0.5$ we switch the positions of source and reference, such that the system receives the reference as the source and vice versa. This has two primary effects on our model. Firstly, during fine-tuning of the underlying language model, the source embeddings are aligned with the target language embedding space resulting in more useful source embeddings. Secondly, it forces the model to treat source and reference as interchangeable inputs, allowing it to handle switching of inputs at inference time without excessively hindering the model's predictive ability. Finally, at inference time we embed source $\boldsymbol{s}$, hypothesis $\boldsymbol{h}$, reference $\boldsymbol{r}$ and the alternative reference $\hat{\boldsymbol{r}}$. These embeddings are then passed to the feed-forward neural network in the following six permutations: $[\boldsymbol{s}; \boldsymbol{h}; \boldsymbol{r}]$, $[\boldsymbol{r}; \boldsymbol{h}; \boldsymbol{s}]$, $[\boldsymbol{s}; \boldsymbol{h}; \hat{\boldsymbol{r}}]$, $[\hat{\boldsymbol{r}}; \boldsymbol{h}; \boldsymbol{s}]$, $[\boldsymbol{r}; \boldsymbol{h}; \hat{\boldsymbol{r}}]$ and $[\hat{\boldsymbol{r}}; \boldsymbol{h}; \boldsymbol{r}]$.

Six passes through the feed-forward gives us six predictions. Final, aggregated scores are achieved by taking the mean of the six predictions and multiplying it by 1 minus the standard deviation (σ). The intuition being that $1 - \sigma$ gives something of an idea of confidence of the model at the segment-level and that scaling the mean prediction to penalize lower confidence might align better with human judgement.

10 Experimental Results

Below we present results of our various COMET models on WMT19 evaluation sets as described

Table 3: Kendall's Tau (τ) correlation and standard deviation (σ) across all language pairs for the top 5 high-performing systems.

Model	Avg. Kendall (all)	Avg. Kendall (en)
BLEU	0.387±0.366	0.257±0.395
CHRF	0.387±0.463	0.343±0.513
BERTSCORE (F1)	0.453±0.267	0.429±0.279
BLEURT	-	0.571±0.355
PRISM	0.52±0.270	0.514±0.279
COMET-MQM (large)	0.587±0.277	0.543±0.276
COMET-HTER (large)	0.547±0.325	0.486±0.363
COMET-DA (large)	**0.653±0.233**	**0.629±0.269**
COMET-RANK (base)	0.547±0.256	0.543±0.276

above. Segment-level and document-level results are outlined in the corresponding tables within the body of the paper, the remaining tables for other task results are contained in the Appendices hereto.

10.1 Segment-level Task

Our segment-level results on the shared task test sets for WMT19 are detailed in tables 1 and 2. We note that for all language pairs out of English (Table 1) both our DA Estimator and our COMET-RANK (base) outperform prior metrics, in some cases by a significant margins. The same can be said in most language pairs into English, where we consistently perform at the level competitive with or exceeding prior metric performance in this task. Table 6 (contained in the appendices) further illustrates performance of our models on non-English language pairs. We note that in all settings our COMET models outperform state-of-the-art for these language pairs.

10.2 System-level Task

System-level results are outlined in tables 7, 8 and 9 in the appendix. In most language pairs we outperform the best metrics in terms of correlation with human judgement. For those language pairs for which our metrics are outperformed by others, we note that ours are at least competitive with other, recent metrics.

An unexpected result is that at system-level our COMET-RANK (base) does not perform as well as our Estimators, regardless of its strong segment-level results. We believe that this is an artifact of training directly on DARR data. Since in WMT shared tasks, the DA rating scale employed is defined at the 0-25-50-75-100 point margins, the minimum required difference between two hypothesis to produce DARR judgement is 25 points (Ma et al., 2019). All other segments are discarded, as within that range the notion of which hypothesis is better becomes ambiguous. As a result we believe that our ranker model learns to successfully discriminate less ambiguous examples and struggles to correctly assign a score otherwise.

10.2.1 Robustness to high-performing systems

As outlined above, we also complement our evaluation at system-level with an analysis of metric performance in terms of the pairwise ranking of the top five performing systems from each language pair. For each setting we output the Kendall's Tau (that is to say the formulation outlined in section 5 above) and report the mean and standard deviation of results across language pairs in table 3.

In both settings we note that our DA Estimator (large) model significantly outperforms other metrics both in terms of mean and standard deviation. This strongly suggests that not only do we perform well in terms of system-level Pearson but that at a practical level, our model can much more successfully differentiate high-performing systems.

10.3 Document-level Task

Table 10 compares the micro-averaging against a simple unweighted average. From table 10 we can observe that micro-averaging outperforms macro-averaging by a small margin. Table 4 summarizes our results for the Document-level Task using our segment-level Estimators with micro-averaging. In this task, the HTER Estimator shows generally superior performance on average surpassing our best performing segment-level model, the DA Estimator. An important conclusion to draw from the strong document-level correlations noted here is that a

model trained to generate segment-level scores, can also perform well as a document-level metric.

Table 4: Pearson correlation (r) between Document-level DAs and micro average segment-level scores for English-to-German and English-to-Czech.

	en-cs	**en-de**	
Nº Documents	1115	2355	**avg.**
COMET-MQM (large)	0.638	0.516	0.577
COMET-HTER (large)	0.655	**0.558**	**0.607**
COMET-DA (large)	**0.667**	0.528	0.598

Table 5: Pearson correlations (r) and adequacy (as reported in Freitag et al. (2020)) for segment-level DA using our DA Estimator (large) model on WMT19 Metrics shared task test data for en-de. We show Pearson's r for the single reference scenario using the corresponding reference ('1-ref') and the multi-reference scenario where the reference is combined with the original in the manner outlined in section 9 above ('2-ref').

Reference	**Adequacy**	r **(1-ref)**	r **(2-ref)**
WMT	85.3	0.523	-
AR	86.7	0.539	0.555
WMTp	81.8	0.470	0.529
ARp	80.8	0.476	0.537

10.4 Multi-Reference Handling

Additional references were obtained for two language pairs: en-de and de-en. For the former, we conducted experiments using 3 additional references from Freitag et al. (2020): AR reference (an additional high quality reference translation), ARp reference (a paraphrased-as-much-as-possible version of AR), and WMTp reference (a paraphrased-as-much-as-possible version of the original WMT reference); for the latter, we use the alternative reference given in the WMT19 News shared task test set. Conveniently, Freitag et al. (2020) also offer a notion of the quality of the extra references for en-de by providing human-generated adequacy assessments for each. In table 5 we show the performance of our DA Estimator (large) model with each reference, either as a single reference or combined in the manner described in section 9 above with the original reference.

While we lack data to draw any statistically significant conclusions, there is a strong suggestion from these results of a positive correlation between reference quality and utility to the predictive model.

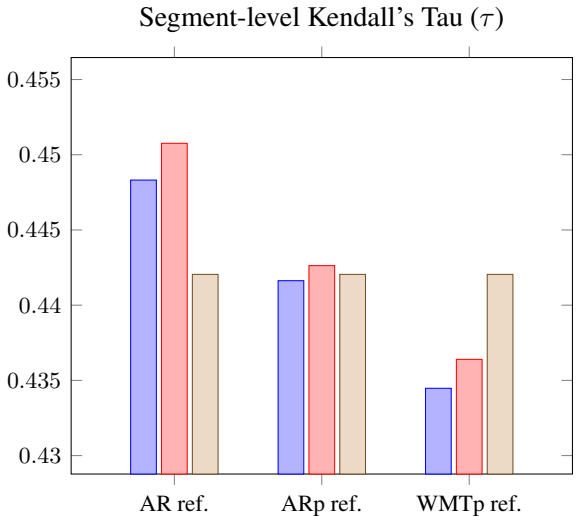


Figure 1: Performance impact of using different kinds of references in combination with the original WMT English-to-German reference. In — we observe the Kendall-Tau τ ranking correlation achieved by our multi-reference Estimator model (section 9). In — we present the Kendall-Tau τ ranking correlation of our "one-reference" Estimator model using the alternative reference. Finally, for comparison, in — we show the Kendall-Tau τ ranking correlation of our "one-reference" Estimator model using the original reference.

For de-en, using an alternative reference did not offer any gain in Pearson's r. We note that when using it alone we only achieve r=0.34 compared to using the original reference which achieves r=0.42. We speculate, based on our observations above, that this might be due to the alternative reference being of lower quality.

These results potentially show that for approaches such as COMET, quality is more important than quantity, and that lower-quality additional references can potentially hurt rather than help improve the correlations obtained using only one single high-quality reference.

With regard to the Kendall Tau measured at segment-level, by looking at Figure 1 (en-de), we see no significant differences in using the multi-reference technique. This suggests that having a higher Pearson's r score does not necessarily guarantee a better Kendall's Tau.

We note that by design, with an approach such as COMET that is based on a meaning-representation of references, extra references are expected to provide only minor additional value, especially versus lexical-based metrics such as BLEU (Papineni et al., 2002). Whereas the adequacy of the reference(s)

is (again by design) expected to have a more significant impact on the performance of the model. Our initial results seem to strongly support this hypothesis.

11 Conclusions

In this paper we present COMET, Unbabel's contribution to the WMT 2020 Metrics shared task. We leverage the framework outlined in Rei et al. (2020) to demonstrate state-of-the-art or otherwise competitive levels of correlation with human judgements in all tasks and introduce a novel method of making optimal use of alternative references and demonstrate that the quality of the reference used is relevant to the success of our framework. Further investigation of the latter, in particular how to better leverage different kinds of references, represent an interesting direction for future work.

12 Acknowledgments

We are grateful to Fabio Kepler, Daan Van Stigt, Miguel Vera, and the reviewers, for their valuable feedback and discussions. This work was supported in part by the P2020 Program through projects MAIA and Unbabel4EU, supervised by ANI under contract numbers 045909 and 042671, respectively.

References

Loïc Barrault, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. 2019. Findings of the 2019 conference on machine translation (WMT19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 1–61, Florence, Italy. Association for Computational Linguistics.

Aljoscha Burchardt and Arle Lommel. 2014. Practical Guidelines for the Use of MQM in Scientific Research on Translation quality. *Quality Translation 21*. (access date: 2020-05-26).

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Markus Freitag, David Grangier, and Isaac Caswell. 2020. BLEU might be Guilty but References are not Innocent. *ArXiv*, abs/2004.06063.

Yvette Graham, Timothy Baldwin, Alistair Moffat, and Justin Zobel. 2013. Continuous measurement scales in human evaluation of machine translation. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 33–41, Sofia, Bulgaria. Association for Computational Linguistics.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

Julia Ive, Lucia Specia, Sara Szoc, Tom Vanallemeersch, Joachim Van den Bogaert, Eduardo Farah, Christine Maroti, Artur Ventura, and Maxim Khalilov. 2020. A post-editing dataset in the legal domain: Do we underestimate neural machine translation quality? In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 3692–3697, Marseille, France. European Language Resources Association.

Fabio Kepler, Jonay Trénous, Marcos Treviso, Miguel Vera, António Góis, M. Amin Farajian, António V. Lopes, and André F. T. Martins. 2019. Unbabel's participation in the WMT19 translation quality estimation shared task. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 78–84, Florence, Italy. Association for Computational Linguistics.

Arle Lommel, Aljoscha Burchardt, and Hans Uszkoreit. 2014. Multidimensional Quality Metrics (MQM): A framework for declaring and describing translation quality metrics. *Tradumàtica: tecnologies de la traducció*, 0:455–463.

Qingsong Ma, Johnny Wei, Ondřej Bojar, and Yvette Graham. 2019. Results of the WMT19 metrics shared task: Segment-level and strong MT systems pose big challenges. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 62–90, Florence, Italy. Association for Computational Linguistics.

Nitika Mathur, Timothy Baldwin, and Trevor Cohn. 2020. Tangled up in BLEU: Reevaluating the evaluation of automatic machine translation evaluation metrics. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4984–4997, Online. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of*

the 40th Annual Meeting of the Association for Computational Linguistics, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A Neural Framework for MT Evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Hiroki Shimanaka, Tomoyuki Kajiwara, and Mamoru Komachi. 2019. Machine Translation Evaluation with BERT Regressor. *arXiv preprint arXiv:1907.12679*.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *In Proceedings of Association for Machine Translation in the Americas*, pages 223–231.

Lucia Specia, Kim Harris, Frédéric Blain, Aljoscha Burchardt, Viviven Macketanz, Inguna Skadina, Matteo Negri, , and Marco Turchi. 2017. Translation quality and productivity: A study on rich morphology languages. In *Machine Translation Summit XVI*, pages 55–71, Nagoya, Japan.

Kosuke Takahashi, Katsuhito Sudoh, and Satoshi Nakamura. 2020. Automatic machine translation evaluation using source language inputs and cross-lingual language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3553–3558, Online. Association for Computational Linguistics.

Brian Thompson and Matt Post. 2020. Automatic machine translation evaluation in many languages via zero-shot paraphrasing. *ArXiv*, abs/2004.14564.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating text generation with BERT. In *International Conference on Learning Representations*.

A Appendix A

Table 6: Segment-level Kendall's Tau (τ) correlations on language pairs not involving English for the WMT19 Metrics DARR corpus. COMET-RANK (base) scores are to be replaced with results of the large model.

	de-cs	**de-fr**	**fr-de**	
Nº Tuples	23194	4862	1369	**avg.**
BLEU	0.222	0.226	0.173	0.207
CHRF	0.341	0.287	0.274	0.301
BERTSCORE (F1)	0.356	0.330	0.277	0.321
PRISM	0.452	0.443	**0.421**	0.439
COMET-MQM (large)	0.413	0.422	0.327	0.387
COMET-HTER (large)	0.425	0.449	0.381	0.418
COMET-DA (large)	**0.471**	**0.469**	**0.420**	**0.453**
COMET-RANK (base)	0.389	0.444	0.331	0.388

Table 7: System-level Pearson correlation (r) for the from-English language pairs from WMT19 DA corpus. DARR Ranker (base) scores are to be replaced with results of the large model.

	en-cs	**en-de**	**en-fi**	**en-gu**	**en-kk**	**en-lt**	**en-ru**	**en-zh**	
Nº Systems	11	22	12	11	10	12	12	12	**avg.**
BLEU	**0.988**	0.952	0.978	0.780	0.864	0.979	0.973	0.762	0.910
CHRF	**0.986**	0.983	**0.988**	0.839	0.969	0.964	0.979	0.822	0.941
BERTSCORE (F1)	0.983	0.990	0.969	0.907	0.983	0.972	**0.989**	0.927	0.965
PRISM	0.964	0.987	0.947	-	0.978	0.929	0.914	0.900	0.946
COMET-MQM (large)	0.943	0.968	0.949	0.946	0.979	**0.985**	0.966	0.958	0.962
COMET-HTER (large)	0.948	**0.991**	0.959	0.948	0.965	**0.982**	0.973	0.943	0.964
COMET-DA (large)	0.964	**0.995**	0.969	**0.964**	**0.989**	**0.982**	**0.987**	**0.969**	**0.977**
COMET-RANK (base)	0.943	0.937	0.914	0.817	0.963	0.973	0.861	0.942	0.919

Table 8: System-level Pearson correlation (r) for the into-English language pairs from WMT19 DA corpus. DARR Ranker (base) scores are to be replaced with results of the large model.

	de-en	**fi-en**	**gu-en**	**kk-en**	**lt-en**	**ru-en**	**zh-en**	
Nº Systems	16	11	9	7	11	13	15	**avg.**
BLEU	0.879	0.984	0.975	0.959	0.969	0.840	0.895	0.929
CHRF	0.916	**0.988**	0.967	0.982	0.938	0.942	0.952	0.955
BERTSCORE (F1)	0.949	0.984	**0.990**	**0.995**	0.961	0.901	0.982	0.966
BLEURT (large-512)	0.939	0.984	0.989	0.989	**0.992**	**0.980**	**0.994**	**0.981**
PRISM	**0.954**	0.981	**0.992**	**0.992**	**0.994**	0.905	**0.992**	0.973
COMET-MQM (large)	0.926	0.974	0.972	0.971	0.986	0.889	0.959	0.954
COMET-HTER (large)	0.918	0.953	0.958	0.951	0.983	0.924	0.978	0.952
COMET-DA (large)	0.946	0.983	**0.993**	**0.996**	**0.993**	0.970	**0.993**	**0.982**
COMET-RANK (base)	0.922	0.981	0.963	0.932	0.987	0.674	0.967	0.918

Table 9: System-level Pearson correlation (r) for language pairs not involving English from WMT19 DA corpus.

	de-cs	**de-fr**	**fr-de**	
N^o Systems	9	11	10	**avg.**
BLEU	0.936	0.934	0.869	0.913
CHRF	**0.994**	0.933	0.908	0.945
BERTSCORE (F1)	0.988	0.953	0.942	0.961
PRISM	0.988	0.924	0.922	0.945
COMET-MQM (large)	0.936	0.950	0.885	0.924
COMET-HTER (large)	0.951	0.901	0.924	0.925
COMET-DA (large)	0.973	**0.972**	**0.954**	**0.966**
COMET-RANK (base)	0.819	0.941	0.927	0.896

Table 10: Document-level Pearson correlation (r) for micro average and macro average for English-to-German and English-to-Czech.

	en-cs		**en-de**	
	Micro-avg.	Macro-avg.	Micro-avg.	Macro-avg.
COMET-DA (large)	**0.667**	0.660	0.528	**0.529**
COMET-MQM (large)	0.638	**0.639**	0.516	**0.519**
COMET-HTER (large)	**0.655**	0.650	**0.558**	0.552
	0.653	0.649	**0.534**	0.533

Learning to Evaluate Translation Beyond English
BLEURT Submissions to the WMT Metrics 2020 Shared Task

Thibault Sellam **Amy Pu*** **Hyung Won Chung†** **Sebastian Gehrmann**
{tsellam, puamy, hwchung, gehrmann}@google.com

Qijun Tan **Markus Freitag** **Dipanjan Das** **Ankur P. Parikh**
{qijuntan, freitag, dipanjand, aparikh}@google.com
Google Research

Abstract

The quality of machine translation systems has dramatically improved over the last decade, and as a result, evaluation has become an increasingly challenging problem. This paper describes our contribution to the WMT 2020 Metrics Shared Task, the main benchmark for automatic evaluation of translation. We make several submissions based on BLEURT, a previously published metric which uses transfer learning. We extend the metric beyond English and evaluate it on 14 language pairs for which fine-tuning data is available, as well as 4 "zero-shot" language pairs, for which we have no labelled examples. Additionally, we focus on English to German and demonstrate how to combine BLEURT's predictions with those of YISI and use alternative reference translations to enhance the performance. Empirical results show that the models achieve competitive results on the WMT Metrics 2019 Shared Task, indicating their promise for the 2020 edition.

1 Introduction

The recent progress in machine translation models has led researchers to question the use of n-gram overlap metrics such as BLEU, which focus solely on surface-level aspects of the generated text, and thus may correlate poorly with human evaluation (Papineni et al., 2002; Lin, 2004; Ma et al., 2019; Mathur et al., 2020; Belz and Reiter, 2006; Callison-Burch et al., 2006). This has led to a surge of interest for more flexible metrics that use machine learning to capture semantic-level information (Celikyilmaz et al., 2020). Popular examples of such metrics include YISI-1 (Lo, 2019), ESIM (Mathur et al., 2019), BERTSCORE (Zhang et al., 2020), the Sentence Mover's Similarity (Zhao et al., 2019; Clark et al., 2019), and BLEURT (Sellam et al., 2020). These metrics utilize contextual embeddings from large models such as BERT (Devlin et al., 2019) which have been shown to capture linguistic information beyond surface-level aspects (Tenney et al., 2019).

The WMT Metrics 2020 Shared Task is the reference benchmark for evaluating these metrics in the context of machine translation. It tests the evaluation of systems that are to-English (X $\rightarrow$ En) and to other languages (X $\rightarrow$ Y), which requires a multilingual approach. An additional challenge for learned metrics is that human ratings are not available for all language pairs, and therefore, the models must use unlabeled data and perform zero-shot generalization.

We describe several learned metrics based on BLEURT (Sellam et al., 2020), originally developed for English data. We first extend BLEURT to the multilingual setup, and show that our approach achieves competitive results on the WMT Metrics 2019 Shared Task.[1] We also present several simple BERT-based baselines, which we submit for analysis. Finally, we focus on English to German and enhance BLEURT's performance by combining its predictions with those of YISI (Lo, 2019) as well as by using alternative references.

2 Background and Notations

Task Reference-based NLG evaluation seeks to assign a score to a triplet of sentences *(input, reference, candidate)*, where *input* is a sentence in the source language, *reference* is a reference translation kept secret at inference time, and *candidate* is a translation produced by an MT system.

[*] Work done during a summer internship. Permanent email address: amy_pu@brown.edu.

[†] Work done as a member of the Google AI Residency Program.

[1] We use the following languages for fine-tuning and/or testing: Chinese, Czech, German, English, Estonian, Finnish, French, Gujarati, Kazakh, Lithuanian, Russian, and Turkish. In addition, we also pre-train on Inuktitut, Japanese, Khmer, Pastho, Polish, Romanian, and Tamil.

Proceedings of the 5th Conference on Machine Translation (WMT), pages 921–927
Online, November 19–20, 2020.

Similar to BLEU (Papineni et al., 2002) and the previous editions of the WMT Metrics shared task, we omit the input and treat the task as a regression problem : we aim to learn a function $\boldsymbol{f} : (\boldsymbol{x}, \tilde{\boldsymbol{x}}) \rightarrow y$ that predicts a quality score y for a candidate sentence $\tilde{\boldsymbol{x}} = (\tilde{x}_1, .., \tilde{x}_p)$ given a reference sentence $\boldsymbol{x} = (x_1, .., x_q)$. The function is supervised on a corpus of N human ratings $\{(\boldsymbol{x}_i, \tilde{\boldsymbol{x}}_i, y_i)\}_{n=1}^{N}$.

BLEURT Most experiments presented in this paper are based on BLEURT, a metric that leverages transfer learning to achieve high accuracy and increase robustness (Sellam et al., 2020). BLEURT is a BERT-based regression model (Devlin et al., 2019). It embeds sentence pairs into a fixed-width vector $\boldsymbol{v}_{\text{BERT}} = \text{BERT}(\boldsymbol{x}, \tilde{\boldsymbol{x}})$ with a pre-trained Transformer, and feeds this vector to a linear layer:

$$\hat{y} = \boldsymbol{f}(\boldsymbol{x}, \tilde{\boldsymbol{x}}) = \boldsymbol{W}\boldsymbol{v}_{\text{BERT}} + \boldsymbol{b}$$

where $\boldsymbol{W}$ and $\boldsymbol{b}$ are the weight matrix and bias vector respectively.

In its original (English) version, BLEURT is trained in three stages. (1) It is initialized from a publicly available BERT checkpoint. (2) The model is then "warmed up" by exposing it to millions of sentence pairs $(\boldsymbol{x}, \tilde{\boldsymbol{x}})$, obtained by randomly perturbing sentences from Wikipedia. During this phase, the model learns to predict a wide range of similarity scores that include existing metrics (BERTSCORE, BLEU, ROUGE), scores from an entailment model, and the likelihood that $\tilde{\boldsymbol{x}}$ was generated from $\boldsymbol{x}$ with a round-trip translation by a given translation model. We denote this stage as *mid-training*. (3) In the final stage, the model is fine-tuned on human ratings from WMT Metrics (Bojar et al., 2017; Ma et al., 2018, 2019), using a regression loss $\ell_{\text{supervised}} = \frac{1}{N}\sum_{n=1}^{N} \|y_i - \hat{y}\|^2$. We found that English BLEURT achieved competitive performance on four academic datasets, WebNLG (Gardent et al., 2017), and the WMT Metrics Shared Task years 2017 to 2019.

3 Extending BLEURT Beyond English

3.1 Modeling

An approach to extend BLEURT would be to use MBERT, the public version of BERT pre-trained on 104 languages, and "mid-train" with non-English signals as described above. Yet, the evidence we gathered from early experiments were inconclusive. On the other hand, we did observe that models trained on several languages were often more accurate than monolingual models, possibly due to the larger amount of fine-tuning data. Thus, we opted for a simpler approach where we start with a multilingual BERT model and fine-tune it on all the human ratings data available for all languages (`X` $\rightarrow$ `Y` and `X` $\rightarrow$ `En`). In most cases, we found that such models could perform zero-shot evaluation: if a language `Y` does not have human ratings data, the metric can still perform evaluation in this target language as long as the base multilingual BERT model contains unlabeled data for `Y`, as observed in the past literature (Karthikeyan et al., 2019; Pires et al., 2019).

We experiment with two pre-trained multilingual models: MBERT and MBERT-WMT, a custom multilingual variant of BERT. The MBERT-WMT model is larger that MBERT (24 Transformer layers instead of 12), and it was pre-trained on 19 languages of the WMT Metrics shared task 2015 to 2020.

Details of MBERT-WMT pre-training We trained MBERT-WMT model with an MLM loss (Devlin et al., 2019), using a combination of public datasets: Wikipedia, the WMT 2019 News Crawl (Barrault et al.), the C4 variant of Common Crawl (Raffel et al., 2020), OPUS (Tiedemann, 2012), Nunavut Hansard (Joanis et al., 2020), WikiTitles[2], and ParaCrawl (Esplà-Gomis et al., 2019). We trained a new WordPiece vocabulary (Schuster and Nakajima, 2012; Wu et al., 2016), since the original vocabulary of mBERT does not support the alphabets of Pashto, Khmer and Inuktitut. The model was trained for 1 million steps with the LAMB optimizer (You et al., 2020), using the learning rate 0.0018 and batch size 4096 on 64 TPU v3 chips.

3.2 Experimental Setup

Datasets At the time of writing, no human ratings data is available for WMT Metrics 2020. Therefore, we use the human ratings from WMT Metrics years 2015 to 2019 for both training and evaluation. We do so in two stages. In the first stage, we use 2015 to 2018 for training (216,541 sentence pairs in 8 languages), setting 10% aside for early stopping. We use 2019 as a development set, to choose hyper-parameters and to

[2] https://linguatools.org/tools/corpora/wikipedia-parallel-titles-corpora/

support high-level modeling decisions. In the second stage, we use 2015 to 2019, that is, all the data available, for training and uniformly sample 10% of the data for early stopping and hyper-parameter tuning. This adds 289,895 sentence pairs and 4 additional languages to our training set, approximately doubling the size of the training data. We report our results on the first setup, but submit our predictions to the shared task using the second setup.

Hyper-parameters We run grid search on the learning rate and export the best model, using values {5e-6, 8e-6, 9e-6, 1e-5, 2e-5, 3e-5}. We use batch size 32 and evaluate the model every 1,000 steps on a 10% held-out data set to prevent over-fitting. During preliminary experiments, we additionally experimented with the batch size, dropout rate, frequency of continuous evaluation, balance of languages, pre-training schemes, WordPiece vocabularies, and model architecture.

3.3 Additional Models and Baselines

English BLEURT We fine-tune a new BLEURT checkpoint, following the methodology described above. The main difference with Sellam et al. (2020) is that we incorporate the to-English ratings of year 2019, which were not previously available.

Monolingual baselines based on BERT We experiment with three baselines and submit the results to the WMT Metrics Shared Task for analysis. BERT-L2-BASE and BERT-L2-LARGE are two regression models based on BERT and trained on to-English ratings. We use the same setup as English BLEURT, but we omit the mid-training phase. A similar approach was described in Shimanaka et al. (2019). BERT-CHINESE-L2 is similar to BERT-L2-BASE, but it uses BERT-CHINESE and it is fine-tuned on to-Chinese ratings.

Other Systems We compare our setups to other state-of-the-art learned metrics: BERTSCORE (Zhang et al., 2020), and Yisi (Lo, 2019) all apply rules on top of BERT embeddings while ESIM (Mathur et al., 2019) is a neural sentence similarity model. PRISM (Thompson and Post, 2020) trains a multilingual translation model that is used as a zero-shot paraphrasing system. All the aforementioned systems take sentences pairs as input. Concurrent work has investigated incorporating the source with great success (Rei et al., 2020). We leave this line of research for future work.

4 Results

Tables 1 and 2 show the results in the X → En direction, at the segment- and system-level respectively. In the majority of cases, one of the BLEURT configurations yields the strongest results. The original BLEURT metric seems to perform better at the segment-level. At the system-level it may be dominated by PRISM (3 out of 7 language pairs) or by one of the simpler BERT-based models (4 out of 7 language pairs).

Tables 3 and 4 present the results for the other languages. MBERT-WMT yields solid results at the segment-level (it achieves the highest correlations for 7 out of 11 language pairs), in particular for the "zero-shot" setups, En → Gu, En → Kk, and En → Lt. It outperforms MBERT consistently, except for En → Ru and En → Zh where it lags behind the other metrics. The results are consistent at the system-level.

Strategy for the WMT Metrics Shared Task Based on these results, we make two "competitive" submissions. We present BLEURT as described above, which we ran on all the X → En sentence pairs. Additionally, we submitted a multilingual system that combines MBERT-WMT (for all languages except Chinese) and BERT-CHINESE-L2 (for Chinese). We ran the multilingual system for all language pairs including to-English, as the large amount of non-English fine-tuning data made available in 2019 may benefit this setup too. We also release the predictions of BERT-BASE-L2, BERT-LARGE-L2, and MBERT for analysis.

5 Additional Improvements on English→German

For English→German, the organizers of WMT20 provide three different reference translations: two standard references and one additional paraphrased reference. Given this novel setup, we investigate how to combine our predictions. Moreover, we use a similar framework to ensemble the predictions of different metrics. In particular, we average the predictions of BLEURT, YISI-1 and YISI-2. All three metrics are different in their approaches. While BLEURT and YISI-1 are reference-based metrics, YISI-2 is reference-

	de-en	fi-en	gu-en	kk-en	lt-en	ru-en	zh-en	**avg**
YiSi	0.164	0.347	0.312	0.440	0.376	0.217	0.426	0.326
YiSi1-SRL	0.199	0.346	0.306	0.442	0.380	0.222	0.431	0.332
ESIM	0.167	0.337	0.303	0.435	0.359	0.201	0.396	0.314
BERTscore	0.176	0.345	**0.320**	0.432	0.381	0.223	0.430	0.330
PRISM	**0.204**	0.357	0.313	0.434	0.382	0.225	0.438	0.336
BLEURT Configurations, English-only								
BERT-L2-base	0.142	0.326	0.274	0.406	0.367	0.197	0.358	0.296
BERT-L2-large	0.172	0.361	0.305	0.424	0.388	0.210	0.420	0.326
BLEURT	0.175	**0.365**	0.316	**0.451**	0.397	0.223	**0.444**	**0.339**
BLEURT Configurations, Multi-lingual								
mBERT	0.172	0.352	0.300	0.430	0.388	0.222	0.397	0.323
mBERT-WMT	0.187	0.363	0.306	0.439	**0.398**	**0.226**	0.425	0.335

Table 1: Segment-level agreement with human ratings on the WMT19 Metrics Shared Task on the to-English language pairs. The metric is WMT's Direct Assessment metric, a robust variant of Kendall τ. The scores for YiSi, YiSi1-SRL, and ESIM come from Ma et al. (2019). The scores for BERTscore and PRISM come from Thompson and Post (2020).

	de-en	fi-en	gu-en	kk-en	lt-en	ru-en	zh-en	**avg**
YiSi	0.949	0.989	0.924	0.994	0.981	0.979	0.979	0.971
YiSi1-SRL	0.950	0.989	0.918	0.994	0.983	0.978	0.977	0.969
ESIM	0.941	0.971	0.885	0.986	0.989	0.968	0.988	0.961
BERTscore	0.949	0.987	0.981	0.980	0.962	0.921	0.983	0.966
PRISM	**0.954**	0.983	0.764	**0.998**	**0.995**	0.914	0.992	0.943
BLEURT Configurations, English-only								
BERT-L2-base	0.938	**0.992**	**0.930**	0.992	0.991	0.976	**0.997**	**0.974**
BERT-L2-large	0.940	0.987	0.819	0.992	0.990	**0.985**	0.993	0.958
BLEURT	0.943	0.989	0.865	0.996	**0.995**	0.984	0.990	0.966
BLEURT Configurations, Multi-lingual								
mBERT	0.937	0.976	0.863	0.984	0.978	0.959	0.978	0.954
mBERT-WMT	0.950	0.991	0.815	0.989	0.992	0.968	0.980	0.955

Table 2: System-level agreement with human ratings on the WMT19 Metrics Shared Task on the to-English language pairs. The metric is Pearson's correlation. The scores for YiSi, YiSi1-SRL, and ESIM come from Ma et al. (2019). The scores for BERTscore and PRISM come from Thompson and Post (2020).

free and calculates its score by comparing translations only to the source sentence. BLEURT is fine-tuned on previous human ratings, while YiSi-1 is based on the cosine similarity between BERT embeddings of the reference and the candidate.

In the remainder of this section, we report BLEURT results using the mBERT-WMT setup unless specified otherwise.[3]

5.1 Modifications to YiSi-1

Before combining BLEURT and YiSi, we perform a series of modifications to YiSi-1 and evaluate their impact on English→German.

Experimental Setup All experimental results are summarized in Table 5. We report both segment-level (DARR) and system-level (Kendall τ) correlations. To replicate the multi-reference setup of 2020, we compute correlations with the standard WMT references as well as the paraphrased reference from Freitag et al. (2020).

Improving YiSi's Predictions Our baseline is similar to the YiSi-1 submission from WMT 2019 (Lo, 2019): we run YiSi-1 with the public multilingual mBERT checkpoint. We then experiment with the underlying checkpoint. We continued pre-training mBERT on the in-domain German NewsCrawl dataset. The resulting model *+pre-train NewsCrawl layer 9* increases the correlation for both reference translations. We improve the correlation further on the paraphrased reference by using the 8th instead of the 9th layer.

Other experiments We tried pre-training BERT on forward translated sentences from German NewsCrawl, to adapt the word embeddings to MT outputs. We also trained a BERT model from scratch on the German NewsCrawl data. These experiments did not result in higher correlations with human ratings.

[3]We use a different checkpoint from the one described in Section 4. The model was trained for 880K steps instead of 1 million, and it uses a sequence length of 256 tokens instead of 128.

	en-cs	en-de	en-fi	*en-gu*	*en-kk*	*en-lt*	en-ru	en-zh	de-cs	*de-fr*	fr-de	**avg**
YiSi1	0.475	0.351	0.537	0.551	0.546	0.470	0.585	0.355	0.376	0.349	0.310	0.446
YiSi1-SRL	-	0.368	-	-	-	-	-	0.361	-	-	0.299	-
ESIM	-	0.329	0.511	-	0.510	0.428	0.572	0.339	0.331	0.290	0.289	-
BERTSCORE	0.485	0.345	0.524	0.558	0.533	0.463	0.580	0.347	0.352	0.325	0.274	0.435
PRISM	0.582	**0.426**	0.591	0.313	0.531	0.558	0.584	0.376	0.458	**0.453**	0.426	0.482
BLEURT Configurations												
BERT-CHINESE-L2	-	-	-	-	-	-	-	0.356	-	-	-	-
MBERT	0.506	0.364	0.551	0.550	0.529	0.516	**0.592**	**0.381**	0.385	0.388	0.291	0.459
MBERT-WMT	**0.603**	0.422	**0.615**	**0.577**	**0.558**	**0.584**	0.492	0.337	**0.461**	0.449	**0.427**	**0.502**

Table 3: Segment-level agreement with human ratings on the WMT19 Metrics Shared Task on non-English language pairs. The metric is WMT's Direct Assessment metric, a robust variant of Kendall τ. Languages without fine-tuning data are denoted in *italics*. The scores for YiSi, YiSi1-SRL, and ESIM come from Ma et al. (2019). The scores for BERTSCORE and PRISM come from Thompson and Post (2020).

	en-cs	en-de	en-fi	*en-gu*	*en-kk*	*en-lt*	en-ru	en-zh	de-cs	*de-fr*	fr-de	**avg**
YiSi1	0.962	**0.991**	0.971	0.909	0.985	0.963	**0.992**	0.951	0.973	0.969	0.908	0.961
YiSi1-SRL	-	**0.991**	-	-	-	-	-	0.948	-	-	0.912	-
ESIM	-	**0.991**	0.957	-	0.980	**0.989**	0.989	0.931	0.980	0.950	0.942	-
BERTSCORE	0.981	0.990	0.970	0.922	0.981	0.978	0.989	0.925	0.969	0.971	0.899	0.961
PRISM	0.958	0.988	0.949	0.624	0.978	0.937	0.918	0.898	0.976	0.936	0.911	0.916
BLEURT Configurations												
BERT-CHINESE-L2	-	-	-	-	-	-	-	**0.953**	-	-	-	-
MBERT	0.942	0.987	0.953	0.949	0.982	0.950	0.947	0.949	0.972	0.970	0.924	0.957
MBERT-WMT	**0.993**	**0.991**	**0.987**	**0.959**	**0.993**	**0.989**	0.888	**0.953**	**0.986**	**0.988**	**0.962**	**0.972**

Table 4: System-level agreement with human ratings on the WMT19 Metrics Shared Task on non-English language pairs. The metric is Pearson's correlation. Languages without finetuning data are denoted in *italics*. The scores for YiSi, YiSi1-SRL, and ESIM come from Ma et al. (2019). The scores for BERTSCORE and PRISM come from Thompson and Post (2020).

Ref	Metric	model	sys-level Kendall τ	seg-level DARR
std	BLEURT	MBERT-WMT¶	**0.896**	**0.420**
std	YiSi-1	MBERT *(WMT19 subm.)*	0.810	0.351
		+pre-train NewsCrawl layer 9	0.870	0.373
		+pre-train NewsCrawl layer 8 †	0.853	0.376
para	BLEURT	MBERT-WMT¶	0.852	**0.413**
para	YiSi-1	MBERT *(WMT19 subm.)*	0.844	0.316
		+pre-train NewsCrawl layer 9	0.887	0.365
		+pre-train NewsCrawl layer 8 †	**0.896**	0.373
src	YiSi-2	MBERT¶	0.307	0.106
2std+para	YiSi-comb	comb of 3 († systems)	**0.905**	0.399
	all-comb	avg of 7 († & ¶ systems)	0.878	**0.454**

Table 5: Agreement with human ratings on the WMT19 Metrics Shared Task for English→German. The first set of results are generated by using the standard reference translations for WMT 2019. The second set of results is generated by using the paraphrased reference translations. YiSi-2 is reference free and only uses the source sentences.

5.2 Combining BLEURT, YiSi-1 and YiSi-2 on Multiple References

We describe our two submissions to WMT 2020, YiSi-COMB and ALL-COMB, which result from our efforts to use multiple references for automatic evaluation. YiSi-COMB is a multi-reference version of the YiSi score (Lo, 2019) aimed at achieving better system-level correlations. ALL-COMB leverages metrics from BLEURT, YiSi-1, and YiSi-2 on multiple references to achieve better segment-level correlation.

YiSi-COMB YiSi scores are F_1 scores of YiSi precision and YiSi recall. For the YiSi-COMB submission, we take the minimum of the YiSi recalls for the three different references as the multi-reference recall, and the maximum of the YiSi precision as the multi-reference precision. Using the same notations as in (Lo, 2019), the final score is the F_1 of the recall and precision computed with $\alpha = 0.7$ (see Figure 1). This submission aims to maximize the system-level correlation.

As shown in Table 5, YiSi-1 has the highest system-level correlation on paraphrased references. Given that we used $\alpha = 0.7$, YiSi scores are quite similar to YiSi recalls (when $\alpha = 1.0$, YiSi scores are equal to YiSi recalls). YiSi-1 scores for paraphrased references are usually much lower than those of standard references, therefore taking the minimum recall is oftentimes equivalent to taking the YiSi recall from the paraphrased references. Furthermore, we found that using the maximum precision, in combination with aggregating recalls, usually performs the best.

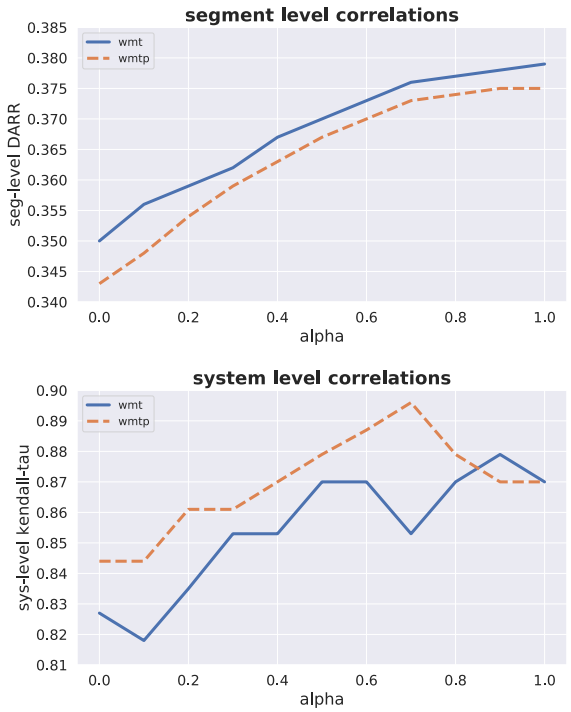


Figure 1: Correlations with respect to different α settings for Yisi-1. The system-level correlation is highest when $\alpha = 0.7$, which is the α we use for the submission.

ALL-COMB We combined the predictions of YISI-1 with those of BLEURT and YISI-2. YISI-2 usually performs worse than the reference-based metrics, but we found that incorporating its predictions can help. Having three different metrics (BLEURT, YISI-1, YISI-2) and three different reference translations, we take all seven predictions and average the scores for each segment. The combined prediction ALL-COMB outperforms every single metric at the segment level, though the system-level correlation drops in comparison to the best YISI-1 score on paraphrased references. This submission aims to maximize the segment-level correlation.

6 Summary

We submit the following systems to the WMT Metrics shared task:

- BLEURT as previously published, fine-tuned on the human ratings of the WMT Metrics shared task 2015 to 2019, to-English.
- A multi-lingual extensions of BLEURT based on a 20 languages variant of MBERT and BERT-CHINESE.
- Three baseline systems based on BERT-BASE, BERT-LARGE, and MBERT.
- Two combination methods for English to German that use YiSi and alternative references, YISI-COMB and ALL-COMB.

7 Acknowledgements

Thanks to Xavier Garcia and Ran Tian for advice and proof-reading.

References

Loïc Barrault, Ondřej Bojar, Marta R Costa-Jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, et al. Findings of the 2019 conference on machine translation. In *Proceedings of WMT*.

Anja Belz and Ehud Reiter. 2006. Comparing automatic and human evaluation of nlg systems. In *Proceedings of EACL*.

Ondřej Bojar, Yvette Graham, and Amir Kamran. 2017. Results of the wmt17 metrics shared task. In *Proceedings of WMT*.

Chris Callison-Burch, Miles Osborne, and Philipp Koehn. 2006. Re-evaluation the role of bleu in machine translation research. In *Proceedings of EACL*.

Asli Celikyilmaz, Elizabeth Clark, and Jianfeng Gao. 2020. Evaluation of text generation: A survey. *arXiv*.

Elizabeth Clark, Asli Celikyilmaz, and Noah A Smith. 2019. Sentence mover's similarity: Automatic evaluation for multi-sentence texts. In *Proceedings of ACL*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL HLT*.

Miquel Esplà-Gomis, Mikel L Forcada, Gema Ramírez-Sánchez, and Hieu Hoang. 2019. Paracrawl: Web-scale parallel corpora for the languages of the eu. In *Proceedings of Machine Translation Summit XVII Volume 2: Translator, Project and User Tracks*.

Markus Freitag, David Grangier, and Isaac Caswell. 2020. BLEU might be Guilty but References are not Innocent. In *Proceedings of EMNLP*.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. The webnlg challenge: Generating text from rdf data. In *Proceedings of INLG*.

Eric Joanis, Rebecca Knowles, Roland Kuhn, Samuel Larkin, Patrick Littell, Chi-kiu Lo, Darlene Stewart, and Jeffrey Micher. 2020. The nunavut hansard inuktitut–english parallel corpus 3.0 with preliminary machine translation results.

Kaliyaperumal Karthikeyan, Zihan Wang, Stephen Mayhew, and Dan Roth. 2019. Cross-lingual ability of multilingual bert: An empirical study. In *Proceedings of ICLR*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Workshop on Text Summarization Branches Out*.

Chi-kiu Lo. 2019. Yisi-a unified semantic mt quality evaluation and estimation metric for languages with different levels of available resources. In *Proceedings of WMT*.

Qingsong Ma, Ondřej Bojar, and Yvette Graham. 2018. Results of the wmt18 metrics shared task: Both characters and embeddings achieve good performance. In *Proceedings of WMT*.

Qingsong Ma, Johnny Wei, Ondřej Bojar, and Yvette Graham. 2019. Results of the wmt19 metrics shared task: Segment-level and strong mt systems pose big challenges. In *Proceedings of WMT*.

Nitika Mathur, Tim Baldwin, and Trevor Cohn. 2020. Tangled up in bleu: Reevaluating the evaluation of automatic machine translation evaluation metrics. *Proceedings of ACL*.

Nitika Mathur, Timothy Baldwin, and Trevor Cohn. 2019. Putting evaluation in context: Contextual embeddings improve machine translation evaluation. In *Proceedings of ACL*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of ACL*.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual bert? *arXiv*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. In *Journal of Machine Learning Research*.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. Comet: A neural framework for mt evaluation. *arXiv*.

Mike Schuster and Kaisuke Nakajima. 2012. Japanese and Korean voice search. In *Proceedings of ICASSP*.

Thibault Sellam, Dipanjan Das, and Ankur P Parikh. 2020. Bleurt: Learning robust metrics for text generation. *Proceedings of ACL*.

Hiroki Shimanaka, Tomoyuki Kajiwara, and Mamoru Komachi. 2019. Machine translation evaluation with bert regressor. *arXiv*.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. Bert rediscovers the classical nlp pipeline. In *Proceedings of ACL*.

Brian Thompson and Matt Post. 2020. Automatic machine translation evaluation in many languages via zero-shot paraphrasing. *Proceedings of EMNLP*.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *Proceedings of The 8th Language Resources and Evaluation Conference*.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. *arXiv*.

Yang You, Jing Li, Sashank Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. 2020. Large batch optimization for deep learning: Training bert in 76 minutes. In *Proceedings of ICLR*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. *Proceedings of ICLR*.

Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M Meyer, and Steffen Eger. 2019. Moverscore: Text generation evaluating with contextualized embeddings and earth mover distance. *Proceedings of EMNLP*.

Towards a Better Evaluation of Metrics for Machine Translation

Peter Stanchev **Weiyue Wang** **Hermann Ney**
Human Language Technology and Pattern Recognition, Computer Science Department
RWTH Aachen University, 52056 Aachen, Germany
<surname>@i6.informatik.rwth-aachen.de

Abstract

An important aspect of machine translation is its evaluation, which can be achieved through the use of a variety of metrics. To compare these metrics, the workshop on statistical machine translation annually evaluates metrics based on their correlation with human judgement. Over the years, methods for measuring correlation with humans have changed, but little research has been performed on what the optimal methods for acquiring human scores are and how human correlation can be measured. In this work, the methods for evaluating metrics at both system- and segment-level are analyzed in detail and their shortcomings are pointed out.

1 Introduction

In the past, machine translation (MT) metrics have been extensively studied and evaluated, at both system- and segment-level (Bojar et al., 2016, 2017; Ma et al., 2018, 2019). When performing system-level evaluation, the average score of a MT system is taken into account. Segment-level evaluation uses each sentence (segment) separately to compute correlation. The results of these metric evaluations are critical to the way MT metrics are perceived. In particular the correlation with human judgment is of great importance.

For this reason, an understanding for the workings of the evaluation method is required. Proposals to identify relevant system-level human scores have been discussed (Koehn, 2012; Sakaguchi et al., 2014), but no comprehensive analysis on this topic has been conducted. In particular, detailed studies on the segment-level evaluation are neglected, although it is an integral part of the metric evaluation.

Since the goal of a metric is to evaluate a translation as close as possible to a human's rating, it is important to clearly define the methods of determining human score and the methods of correlation measurement. This work aims to present an overview of the methods used in the evaluation, analyze their strengths and weaknesses, and propose solutions to some of the pitfalls of the methods.

2 Human Scores

To measure the correlation between the score of a metric and the score of a human, a method of determining human scores is required. Thus, a person has to judge the quality of a translated sentence. This is not a simple task, as different people may have different opinions about the exact quality of the translation. Another aspect to consider is that in order to calculate correlation, the score must be quantifiable in some way. Thus, the methods used to detect human judgment must use a sufficient number of human judges for them to be reproducible.

In the Workshop on Statistical Machine Translation (WMT), three different methods are used to determine the human score: direct assessment (DA) (Graham et al., 2017), relative ranking (RR) (Stanojevic et al., 2015) and, in recent years, relative ranking out of direct assessment (DARR) (Bojar et al., 2017).

2.1 Direct Assessment

The DA measures the quality of a translation on a scale from 0 to 100, based on the adequacy and fluency of the sentence. To obtain the score, the human judges are provided with a reference translation and the output of a single MT system, which makes the evaluation process monolingual. To ensure reproducibility, a large number of judges are needed – at least 15 (Ma et al., 2019). Additionally, scores are standardized (Graham et al., 2017) to eliminate individual distortions, such as judges who only provide high or low scores. Furthermore, a form of quality control is applied to filter out

Proceedings of the 5th Conference on Machine Translation (WMT), pages 928–933
Online, November 19–20, 2020.

judges who exhibit a high variance in comparison to their peers.

Overall, DA is one of the best ways to obtain human judgement. It provides a numerical score that can be easily used in common statistical methods, such as Spearman's ρ (Spearman, 1987) or Pearson's r (Pearson and Galton, 1895), at both the segment- and the system-level. However, to obtain statistically significant correlation measurements and ensure reproducibility, a high number of human scores are required. For the segment-level it is therefore infeasible to obtain DA scores. This leads to the need to use a completely different method for determining human judgements at the segment-level. Another possibility is to establish a relative ranking of the few obtained DA scores (DARR).

2.2 Relative Ranking

The RR produces, as the name implies, a ranking between multiple translations. In WMT, the judges are presented with five system outputs with the corresponding source and reference sentence, making the evaluation process bilingual. Each judge ranks the five sentences from the best to worst, taking equality (tie) into account. To simplify the evaluation, identical sentences from different systems are collapsed into one.

The resulting relative ranking of five tuples is not as straightforward to use for correlation calculation, since most correlation coefficients rely on absolute ranking information. One approach to obtaining a correlation is to use a variant of Kendall's τ (Kendall, 1938; Macháček and Bojar, 2014). This entails converting the scores produced by metrics into relative rankings. Naturally, this has the disadvantage that the fine granularity of the scores is lost. However, this method can be used for both segment- and system-level correlation calculations.

Another option used in WMT16 (Bojar et al., 2016) is to convert relative rankings to absolute rankings through TrueSkill (Herbrich et al., 2006; Sakaguchi et al., 2014). This method uses the relative rankings to estimate an absolute score for each system, which is then used to calculate the correlation (by Pearson's r or Spearman's ρ). The score of each system is represented by a Gaussian distribution, with the mean of the predicted score of the system and the variance of the confidence in that prediction. Due to the nature of the method, it can only be used for the system-level correlation calculation. This, in turn, makes it difficult to interpret the results since normally two different correlation calculation methods must be used for the different evaluation levels.

2.3 DARR

Due to the difficulty of obtaining enough DA scores for a statistically significant segment-level correlation calculation, Bojar et al. (2017) introduced the concept of obtaining a relative ranking from the DA used at the system-level and termed DARR. For this purpose, all possible sentence pairs, for which a DA score is available, are generated between all participating systems. These sentence pairs are then filtered to remove ties. The criterion used by Ma et al. (2019) is to remove sentence pairs, whose difference on the DA scale is less than 25. This should lead to the removal of all ties and produce an RR that scores the systems only as better or worse. However, this is not the case. Table 1 shows the RR of sentences with a sentence identifier (`SID`) on different language pairs (`LP`). The system that has achieved a better translation according to the DA score for these sentences is under the column `better`. In this case, the sentences generated by both systems are completely identical, as can be seen in Table 2, although they have been classified as different according to the DARR method. Such identical sentences occur across multiple language pairs in the WMT19 data set.

Another important aspect is that tie filtering is not applied to the metrics scores and therefore ties are possible for metrics. This makes the correlation calculation, especially for identical sentences, a difficult task. It is therefore of interest to determine how many identical sentences are present after filtering. For this reason, a brief analysis is carried out on the basis of the WMT19 data using six language pairs, which is shown in Table 3. There are no identical sentences for the language pairs Gujarati→English (`gu-en`) and Kazakh→English (`kk-en`). However, for all other language pairs, especially Chinese→English (`zh-en`), there are identical sentences. Note that these identical sentences are present after the tie filtering. Table 3 also shows the amount of ties produced by two metrics: YiSi-1 (Lo, 2019) and EED (Stanchev et al., 2019). It is clear that a significant amount of the ties for the two metrics come from identical sentences.

In addition, a considerable amount of data is eliminated. Figure 1 depicts the effect of varying the equivalence threshold, i.e. cases, in which the

LP	data	SID	better	worse
de-en	newstest2019	1200	uedin.6749	UCAM.6461
zh-en	newstest2019	604	Baidu-system.6940.zh-en	MSRA.MASS.6996.zh-en
en-zh	newstest2019	1351	NEU.6830	UEDIN.6158

Table 1: RR human scores for segment-level with their corresponding sentences from WMT19[1].

LP	SID	system-generated hypotheses (identical for both systems)
de-en	1200	Music cabaret: Gender understanding with heart - Wolbeck - Westphalian News
zh-en	604	China Resources Beer closed at HK $28.85 on Friday, down nearly 4.5% in the past month.
en-zh	1351	他还希望赋予议会更大的权力来建造新的住房。

Table 2: Corresponding sentences from Table 1 for the two systems.

LP	#sentences	#identical sentences	YiSi-1 #ties	EED #ties
gu-en	31k	0	2	27
kk-en	27k	0	74	115
zh-en	31k	152	336	361
en-gu	11k	5	8	13
en-kk	18k	23	53	64
en-zh	19k	84	205	455

Table 3: Number of identical sentences vs. ties in the WMT19 corpus used for human correlation.

difference in DA scores is below the specified value, are considered as ties. Note that the threshold influences the amount of data used immensely. By having virtually no threshold (a threshold of 1) the average number of sentences is five times higher than when using a threshold of 50. The threshold of 25 used by WMT19 almost halves the amount of data used to acquire the correlation.

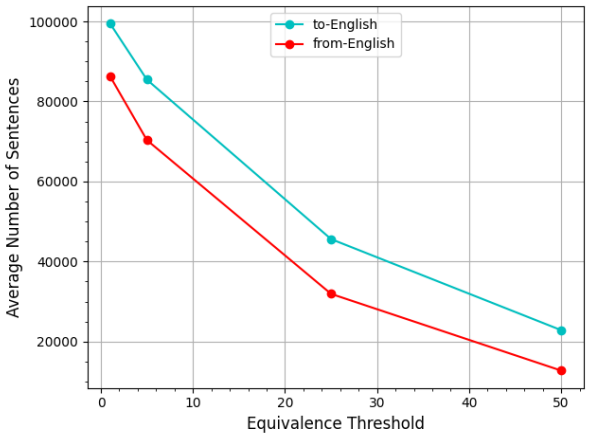


Figure 1: The average number of sentences over different language pairs (`to-English` and `from-English` directions) when excluding ties based on various equivalence thresholds.

Overall DARR provides a method for calculating the correlation at the segment-level in a scenario where there is not enough DA data. However, removing ties as part of the human component makes the evaluation unfair. This is aggravated by the fact that after DA-based tie filtering, not all ties are successfully removed. One possible solution, which remains to be tested, is to consider the ties of the human component carefully. This would at least equal the domain of metrics and human scores.

3 Measuring the Correlation

Obtaining human scores is only part of the correlation calculation. The other one is to use both human and metric scores to compute their similarity or correlation. The case, where both human and metric scores are represented by absolute values, is straightforward to compute using methods such as Pearson's r or Spearman's ρ. However, DA relies on a large amount of annotators that cannot always be guaranteed, especially at the segment-level. In the case where RR or DARR is used for human scores, this task is not that easy. For this reason, the focus here is on the case where a form of RR is used – typically for the segment-level correlation calculation.

As previously mentioned, WMT uses a form of Kendall's τ to obtain a correlation given the relative ranking. The coefficient definition in its most general form is shown in Equation (1)

$$\tau = \frac{|\text{concordant} - \text{discordant}|}{|\text{concordant} + \text{discordant}|}, \qquad (1)$$

where the concordant pairs denote cases in which there is agreement between the metric and the hu-

[1]Scripts and data from: http://ufallab.ms.mff.cuni.cz/~bojar/wmt19-metrics-task-package.tgz

man score, and the discordant pairs cases in which there is disagreement.

To formally define agreement and disagreement, a matrix can be used as described by Macháček and Bojar (2014). The various matrix formulations that have been used in WMT over the years are shown in Table 4. For metric scores to be interpretable in these matrices, a relative ranking must be constructed from the absolute scores for each participating metric. This is achieved by performing a pairwise comparison of the participating systems at the segment-level, taking into account ties. All three matrices treat matches and mismatches identically:

- discordant pairs are always cases where there are disagreements between the human and metric scores: $\{<, >\}$ or $\{>, <\}$,
- concordant pairs are always cases where the scores match: $\{<, <\}$ or $\{>, >\}$.

The only difference between the three methods is the treatment of ties.

Table 4a ignores the existence of ties. However, this is not desirable since ties are possible for metrics. Therefore, metrics are not evaluated on the same amount of sentences. This can be particularly detrimental to metrics that produce a large number of ties. For example, a metric with 99 ties and 1 concordant pair would achieve perfect correlation, while a metric without ties and 70 concordant and 30 discordant pairs would give a correlation of 0.4. The discrepancy in the results due to the data difference is evident.

On the other hand, incorporating ties while not considering human score ties can also lead to undesirable results. In Table 4c, which is used in WMT19, metric ties are considered as a discordant pair $\{<, =\}$ and $\{>, =\}$. Since ties are not defined (or included) in human scores, every tie produced by a metric results in a discordant pair. This in turn reduces its correlation. Thus, a "perfect" metric would never produce a tie between two sentences. This assumption does not reflect reality. In addition, the matrix is not symmetric since there are more possible discordant pairs than concordant ones. This means that a reasonable interpretation of the negative correlation is not possible. Therefore, metrics that have a negative correlation, such as TER (Snover et al., 2006), CHARACTER (Wang et al., 2016) and EED (Stanchev et al., 2019), must be mapped from an error (or edit) rate (E) to an accuracy score to ensure a relatively fair evaluation. This is not trivial, as there is no standard way to convert these metrics into the accuracy rate: neither $1 - E$ nor $-E$ is optimal.

A middle ground between the penalization and the ignoring of ties is the matrix in Table 4b. The ties are not penalized directly, but affect the overall correlation since they are part of the denominator:

$$\tau = \frac{|\text{concordant} - \text{discordant}|}{|\text{concordant} + \text{discordant} + \text{ties}|} \quad (2)$$

Since there is no hard penalization for metrics that produce more ties, such metrics are at a disadvantage. For example, a metric with 20 ties and 80 concordant pairs would achieve a correlation of $80/(80 + 20) = 0.8$, although all non-tie pairs achieve perfect correlation. On the other hand, a metric that overproduces ties, for example, with 80 ties and 20 concordant pairs, would have a correlation of $20/(20 + 80) = 0.2$. It can also be argued that measuring the correlation on metrics with a too high percentage of ties is not significant, since there are too few sentence pairs that are concordant or discordant.

One possible solution to the problem is shown in Table 5. The cases where there is clear agreement or disagreement between humans and metrics remain unchanged. In cases of tie disagreements, a soft penalization is added. This soft penalization is realized the same manner as in Table 4b using Equation (2). In the case where both the metric and human scores tie the two systems, a concordant pair (1) for accuracy-based metrics and a discordant pair (-1) for error rate-based metrics are given. This allows the process to be symmetrical and avoids the problem of having to map error rate to accuracy or vice versa. In addition, ties can now positively affect the correlation and all metrics are evaluated on the same amount of data. Naturally, this alteration of the evaluation method requires that ties be included in the RR. When using DARR, this can be achieved by considering all pairs, where the DA score difference is less than 25, and where the system translations are identical, as ties. A disadvantage of this method is that a distinction has to be made between metrics that aim for a strong negative correlation and metrics that aim for a strong positive correlation. Moreover, the exact range, where a tie is considered, is not necessarily clear.

		Metric		
		<	=	>
Human	<	1	X	-1
	=	X	X	X
	>	-1	X	1

(a) No tie penalization

		Metric		
		<	=	>
Human	<	1	0	-1
	=	X	X	X
	>	-1	0	1

(b) Soft tie penalization

		Metric		
		<	=	>
Human	<	1	-1	-1
	=	X	X	X
	>	-1	-1	1

(c) Hard tie penalization

Table 4: Kendall's τ evaluation matrices (Machácek and Bojar, 2014; Ma et al., 2019).

		Metric		
		<	=	>
Human	<	1	0	-1
	=	0	{1,-1}	0
	>	-1	0	1

Table 5: Integration of human ties in Kendall's τ.

4 Discussion

The MT metric evaluation is an area that needs further investigation. This work gives an overview of the methods used so far and highlights some of their shortcomings. The system-level assessment currently seems to be good, but the evaluation methods at the segment-level still need to be explored (in particular, if there is not enough DA data to directly calculate the correlation at the segment-level):

- It might not be a good idea to rule out tie cases: in theory, there are identical translations and translations of the same quality, and the metrics should be able to give them the same score; in practice, we have shown that excluding all tie cases eliminated a large proportion of the scores collected, which will have a significant impact on the final results. However, it is difficult to clearly define the tie cases for human evaluations, as in DA, on a scale from 0 to 100, different human annotators can give different scores for identical translations.

- The threshold for tie cases is not well defined. Further studies on the threshold value can be carried out. And also whether a threshold should be applied to the automatic metric scores. This study itself may not be a theoretically well-defined task, but some insight could be gained by examining the performance of various metrics under different thresholds.

- The used correlation coefficient is not symmetrical. Then the metrics with negative correlations have to be preprocessed before the evaluation, which can lead to inconsistencies. The proposed solution may also have potential problems as described, but it is worth doing further studies to define a better correlation coefficient.

In general, the task of creating a metric evaluation that is fair and reproducible for all metric types remains to be solved and deserves more attention and study.

Acknowledgements

This work has received funding from the European Research Council (ERC) (under the European Union's Horizon 2020 research and innovation programme, grant agreement No 694537, project "SEQCLAS") and the Deutsche Forschungsgemeinschaft (DFG; grant agreement NE 572/8-1, project "CoreTec"). The work reflects only the authors' views and none of the funding parties is responsible for any use that may be made of the information it contains.

References

Ondřej Bojar, Yvette Graham, and Amir Kamran. 2017. Results of the WMT17 metrics shared task. In *Proceedings of the Second Conference on Machine Translation (WMT)*, pages 489–513, Copenhagen, Denmark.

Ondřej Bojar, Yvette Graham, Amir Kamran, and Milos Stanojevic. 2016. Results of the WMT16 metrics shared task. In *Proceedings of the First Conference on Machine Translation, WMT 2016, colocated with ACL 2016, August 11-12, Berlin, Germany*, pages 199–231. The Association for Computer Linguistics.

Yvette Graham, Timothy Baldwin, Alistair Moffat, and Justin Zobel. 2017. Can machine translation sys-

tems be evaluated by the crowd alone. *Natural Language Engineering*, 23(1):3–30.

Ralf Herbrich, Tom Minka, and Thore Graepel. 2006. TrueSkill™: A bayesian skill rating system. In *Advances in Neural Information Processing Systems 19, Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 4-7, 2006*, pages 569–576. MIT Press.

M. G. Kendall. 1938. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93.

Philipp Koehn. 2012. Simulating human judgment in machine translation evaluation campaigns. In *2012 International Workshop on Spoken Language Translation, IWSLT 2012, Hong Kong, December 6-7, 2012*, pages 179–184. ISCA.

Chi-kiu Lo. 2019. YiSi - a unified semantic mt quality evaluation and estimation metric for languages with different levels of available resources. In *Proceedings of the Fourth Conference on Machine Translation*, pages 706–712, Florence, Italy. Association for Computational Linguistics.

Qingsong Ma, Ondřej Bojar, and Yvette Graham. 2018. Results of the WMT18 metrics shared task: Both characters and embeddings achieve good performance. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers, WMT 2018, Belgium, Brussels, October 31 - November 1, 2018*, pages 671–688. Association for Computational Linguistics.

Qingsong Ma, Johnny Wei, Ondřej Bojar, and Yvette Graham. 2019. Results of the WMT19 metrics shared task: Segment-level and strong MT systems pose big challenges. In *Proceedings of the Fourth Conference on Machine Translation, WMT 2019, Florence, Italy, August 1-2, 2019 - Volume 2: Shared Task Papers, Day 1*, pages 62–90. Association for Computational Linguistics.

Matous Machácek and Ondřej Bojar. 2014. Results of the WMT14 metrics shared task. In *Proceedings of the Ninth Workshop on Statistical Machine Translation, WMT@ACL 2014, June 26-27, 2014, Baltimore, Maryland, USA*, pages 293–301. The Association for Computer Linguistics.

Karl Pearson and Francis Galton. 1895. VII. Note on regression and inheritance in the case of two parents. *Proceedings of the Royal Society of London*, 58(347-352):240–242.

Keisuke Sakaguchi, Matt Post, and Benjamin Van Durme. 2014. Efficient elicitation of annotations for human evaluation of machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation, WMT@ACL 2014, June 26-27, 2014, Baltimore, Maryland, USA*, pages 1–11. The Association for Computer Linguistics.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *In Proceedings of Association for Machine Translation in the Americas*, pages 223–231.

C. Spearman. 1987. The proof and measurement of association between two things. *The American Journal of Psychology*, 100(3/4):441–471.

Peter Stanchev, Weiyue Wang, and Hermann Ney. 2019. EED: Extended edit distance measure for machine translation. In *Proceedings of the Fourth Conference on Machine Translation*, pages 713–719, Florence, Italy. Association for Computational Linguistics.

Milos Stanojevic, Amir Kamran, Philipp Koehn, and Ondřej Bojar. 2015. Results of the WMT15 metrics shared task. In *Proceedings of the Tenth Workshop on Statistical Machine Translation, WMT@EMNLP 2015, 17-18 September 2015, Lisbon, Portugal*, pages 256–273. The Association for Computer Linguistics.

Weiyue Wang, Jan-Thorsten Peter, Hendrik Rosendahl, and Hermann Ney. 2016. CharacTer: Translation edit rate on character level. In *Proceedings of the First Conference on Machine Translation, WMT 2016, colocated with ACL 2016, August 11-12, Berlin, Germany*, pages 505–510. The Association for Computer Linguistics.

Incorporate Semantic Structures into Machine Translation Evaluation via UCCA

Jin Xu[1], **Yinuo Guo**[2], **Junfeng Hu**[2]
[1]Yuanpei College, Peking University
[2]Key Laboratory of Computational Linguistics, School of EECS, Peking University
{jinxu, gyn0806, hujf}@pku.edu.cn

Abstract

Copying mechanism has been commonly used in neural paraphrasing networks and other text generation tasks, in which some important words in the input sequence are preserved in the output sequence. Similarly, in machine translation, we notice that there are certain words or phrases appearing in all good translations of one source text, and these words tend to convey important semantic information. Therefore, in this work, we define words carrying important semantic meanings in sentences as *semantic core words*. Moreover, we propose an MT evaluation approach named *Semantically Weighted Sentence Similarity (SWSS)*. It leverages the power of UCCA to identify semantic core words, and then calculates sentence similarity scores on the overlap of semantic core words. Experimental results show that SWSS can consistently improve the performance of popular MT evaluation metrics which are based on lexical similarity.

1 Introduction

Machine Translation Evaluation (MTE) is to evaluate the quality of sentences produced by Machine Translation (MT) systems. Most automatic MT evaluation metrics compare the candidate sentences from MT systems with reference sentences from human translation to produce a similarity score, in contrast some other reference-less metrics directly compare candidate sentences and source sentences.

According to the observation of well-translated sentences, we find out that there are certain words or phrases appearing in all good translations of one source text. This phenomenon is consistent with the intuition of copying mechanism (Gu et al., 2016), which has been widely used in lots of text generation tasks. In the field of MT evaluation, Meteor++ (Guo et al., 2018) firstly proposes the concept of *copy knowledge* to define the words with copy property, and it further incorporates the copy knowledge into Meteor (Denkowski and Lavie, 2014) to improve its performance. Specifically, it attempts to find copy words of references and candidate sentences, and uses the overlap of these words to modify the calculation of precision and recall of Meteor. However, Meteor++ uses named entities as an alternative to copy knowledge in its experiments, resulting in a limited range of selected copy words and a slight improvement.

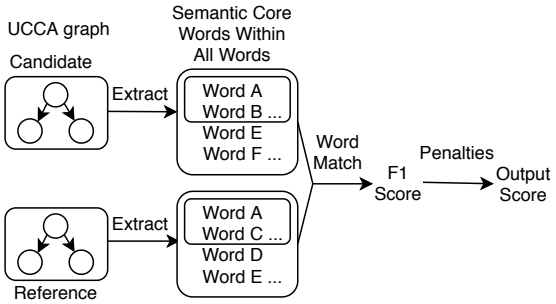


Figure 1: An illustration of the process of SWSS.

In this work, we argue that words undertaking important semantic meanings should be exactly expressed during the translation procedure, which we define as semantic core words. This concept is much more general and closer to linguistic intuition compared to the copy knowledge used in Meteor++. In order to apply semantic core words in the process of MT evaluation, we design a mechanism named *Semantically Weighted Sentence Similarity (SWSS)* illustrated in Figure 1. Firstly, SWSS extracts semantic core words according to the annotated semantic labels in Universal Conceptual Cognitive Annotation (UCCA) (Abend and Rappoport, 2013), a multi-layered semantic representation. UCCA is an appealing candidate for this mechanism as it includes a lot of fundamental semantic phenomena,

Proceedings of the 5th Conference on Machine Translation (WMT), pages 934–939
Online, November 19–20, 2020.

such as verbal, nominal and adjectival argument structures and their inter-relations. Also, semantic units in UCCA are anchored in the text, which simplifies the aligning procedure a lot. With the assumption that all high-quality translations should have the same semantic core words, SWSS then calculates precision and recall based on the overlap of semantic core words between sentence pairs and their corresponding F1 scores. Finally, we modify the F1 score according to the differences of two UCCA representations. For example, *Scenes* are involved in the penalties, which are essential nodes in UCCA indicating actions and states of the sentences. Our experimental results show that SWSS can be combined with other popular MT evaluation metrics to improve their performance significantly.

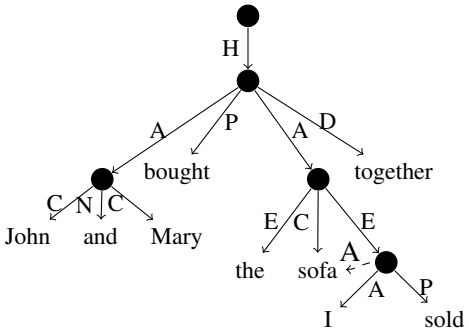


Figure 2: UCCA representation of sentence "John and Mary bought the sofa I sold together". Labels include *Parallel Scene (H)*, *Participant (A)*, *Process (P)*, *Adverbial (D)*, *Center (C)*, *Connector (N)*, *Elaborator (E)*. Dash line indicates a secondary semantic relation. There are two scenes in this sentence, the whole sentence and "I sold (sofa)".

2 Related Work

2.1 Machine Translation Evaluation

BLEU (Papineni et al., 2002) and Meteor are two most popular MT evaluation metrics. BLEU measures n-grams overlapping between the candidate sentences and reference sentences, while Meteor aligns words and phrases to calculate a modified weighted F-score. The two metrics are based on lexical similarity but somehow neglect semantic structure information of the sentences.

Efforts have been made to incorporate linguistic features and resources into MT evaluation. RED (Yu et al., 2014) makes use of dependency tree and MEANT (Lo et al., 2012) makes use of semantic parser. Categories such as part-of-speech (Avramidis et al., 2011) and named entity (Buck, 2012) also have their effects. In order to complement WordNet (Miller, 1998) and paraphrase table in Meteor, Meteor++2.0 (Guo and Hu, 2019) applies syntactic-level paraphrase knowledge.

2.2 Semantic Representation

Semantic representation focuses on how meaning is expressed in a sentence. Some semantic representation frameworks such as UNL (Uchida and Zhu, 2001) and AMR (Banarescu et al., 2013) use concept nodes to represent content words of sentence, and use directed edges with labels to indicate the semantic relation between nodes.

UCCA is a novel multi-layered semantic representation framework, which converts a sentence into a directed acyclic graph (DAG). Leaf nodes of UCCA graph correspond to words in the sentence, and a non-leaf node represents the combination of meanings of its child nodes. A parent node and a child node are connected by a directed edge which demonstrates the semantic role of the child node in the meaning of the parent node. Figure 2 is an example of UCCA representation.

Scene is an essential concept in UCCA. A scene describes some movement, action or a state in the sentence. Scene nodes in UCCA representation may be connected to the root node, or embedded in other scenes as arguments or modifiers. A scene node has a main relation, either a *Process* or a *State*, and may have some *Participants* or some *Adverbials*. These non-scene nodes may also have inner structure.

UCCA has been applied in many fields of Natural Language Processing. SAMSA (Sulem et al., 2018) is a Text Simplification evaluation metric that defines minimal center of UCCA representation and compares simplified text with the minimal centers of original sentences. It is also used in evaluation of faithfulness in Grammatical Error Correlation (Choshen and Abend, 2018) and human MT evaluation (Birch et al., 2016).

3 Proposed Method

3.1 Semantic Core Words

The most popular MT evaluation metrics such as BLEU and Meteor are based on lexical similarity. This kind of metrics cannot obtain insight into semantic structure of the whole sentence. Our proposed semantic core words are extracted from UCCA semantic structures and used to improve

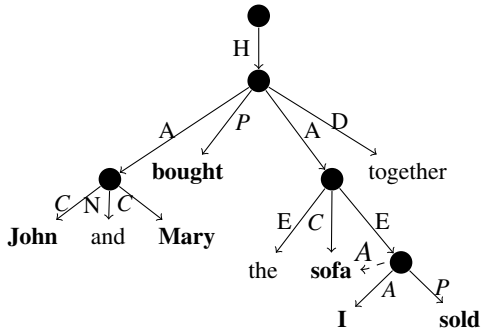


Figure 3: An example of semantic core words. The sentence is the same with Figure 2. All semantic core words are bold and the semantic labels of related edges are italic.

these lexical metrics as we expect them to play the role of copy words.

It is a linguistic intuition that some words carry more semantic information than other words in a sentence. For example, a modifier is usually less important than the word it modifies. In this paper, We define words that have important semantic information as semantic core words. According to their semantic importance, they are expected to be accurately translated during translation. Therefore, we assume that in all good translation results of a specific sentence, the set of semantic core words should be the same, behaving like copy words.

We extract semantic core words of a sentence from its UCCA semantic representation. The lowest semantic role label in the representation for each word is considered, which also indicates the most basic semantic role of a word. A word whose lowest semantic role is *Process*, *State*, *Participant* or *Center* is identified as semantic core words. Figure 3 marks semantic core words of the example sentence. The result is consistent with our intuition of which word has important meaning in this sentence.

3.2 Word Matching

After semantic core words are extracted from UCCA representations, a word matching algorithm should be applied in order to match all words between the two sentences. In this paper, we use a stemming algorithm. Two words are matched if they have the same stem.

We count how many semantic core words in a candidate sentence can be matched to any semantic core words in the reference sentence, and compute the proportion as precision. Similarly, we calculate the matched proportion of semantic core words in reference sentence as recall. In our word matching algorithm, it is possible that a word in a sentence is matched to multiple words in the other sentence because they all have the same word stem. However, just like what is conducted in BLEU, a word cannot be contained in multiple matching pairs. The precision and recall are then used to calculate F1 score. We use F1 score here to ensure that SWSS is symmetrical and can be directly used as a sentence similarity metric.

$$
\begin{aligned}
P &= \frac{\sum_i w(h_i) \cdot m(h_i)}{\sum_i w(h_i)} \\
R &= \frac{\sum_i w(r_i) \cdot m(r_i)}{\sum_i w(r_i)} \\
F_1 &= \frac{2P \cdot R}{P + R}
\end{aligned}
\tag{1}
$$

Take the calculation of precision as an example. h_i means each semantic core word in the candidate sentence, and $w(h_i)$ is its weight. Though in this paper the weight is fixed to 1, it can be fine-tuned or trained in future work. If h_i can be matched to any semantic core word in the reference sentence, $m(h_i)$ is set to 1, otherwise $m(h_i)$ is set to 0. However, $m(h_i)$ can also be different values related to matching type like the operation in Meteor, which might be conducted in future work.

A fact is that the UCCA parser we used might occasionally produce an analysis result in which there are no semantic core words in a sentence, which causes division by zero during calculation. In these cases a fixed score ω is used as an alternative.

3.3 Penalty and Combination

According to the intuition that good translation results of a specific sentence should have similar semantic structures, we introduce three penalties concerning statistical differences of two UCCA representations.

- The ratio between counts of scenes of two representations. Let S_1, S_2 be the counts of scenes, the penalty P_S is $1 - \min(S_1, S_2)/\max(S_1, S_2)$.
- The ratio between counts of nodes of two representations. Let N_1, N_2 be the counts of nodes, the penalty P_N is $1 - \min(N_1, N_2)/\max(N_1, N_2)$.
- The ratio between counts of edges towards critical semantic roles of two representations,

Base Model	BLEU		Meteor		Meteor++	
Method	None	+UCCA	None	+UCCA	None	+UCCA
WMT15						
cs-en	0.377	**0.418**	0.605	**0.609**	0.610	**0.613**
de-en	0.420	**0.464**	0.620	**0.638**	0.637	**0.651**
fi-en	0.378	**0.444**	0.645	**0.668**	0.661	**0.679**
ru-en	0.445	**0.477**	0.628	**0.634**	0.620	**0.629**
Average	0.405	**0.451**	0.624	**0.637**	0.632	**0.643**
WMT16						
cs-en	0.484	**0.508**	**0.649**	0.646	**0.656**	0.651
de-en	0.367	**0.394**	0.503	**0.520**	0.507	**0.523**
fi-en	0.325	**0.368**	0.537	**0.548**	0.557	**0.564**
ro-en	0.418	**0.451**	0.626	**0.633**	0.625	**0.632**
ru-en	0.377	**0.413**	0.574	**0.578**	0.583	**0.585**
tr-en	0.333	**0.401**	0.609	**0.638**	0.600	**0.628**
Average	0.384	**0.423**	0.583	**0.594**	0.588	**0.597**

Table 1: Segment-level Pearson correlation comparison between base model and the combination of SWSS and base model. The smoothing parameter X of Meteor++ is set to 8, which is used on WMT15 dataset in its paper.

which are *Process*, *State* and *Participant*. This count is the sum of count of scenes and count of all arguments in the sentence. Let E_1, E_2 be the counts of these edges, the penalty P_E is $1 - \min(E_1, E_2)/\max(E_1, E_2)$.

The three penalties are set to 0 if the counts are equal and their upper bound is 1. Additionally, we also notice that the average word count of a sentence pair can act as another penalty Len. Applying the four penalties, the final score is calculated by the equation below. All parameters here are tunable.

$$Score = F_1 \cdot \exp(-\alpha_1 \cdot P_S - \alpha_2 \cdot P_N - \alpha_3 \cdot P_E - \alpha_4 \cdot Len) \quad (2)$$

The SWSS score is calculated independently. Therefore, as a semantic structure-based component, it can be further combined with other MT evaluation metrics to obtain a more accurate evaluation metric. For example, we can obtain a simple weighted model of SWSS and Meteor by tuning the weight β below.

$$SWSS \star Meteor = Meteor + \beta \cdot Score \quad (3)$$

4 Experiments

4.1 Data

SWSS is evaluated on WMT15 (Stanojević et al., 2015) and WMT16 metric task (Bojar et al., 2016) evaluation sets and is tuned on WMT17 metric task (Bojar et al., 2017) evaluation set. The datasets are composed of pairs of system output sentences and reference sentences, and also corresponding human evaluation scores for the output sentences.

α_1	0.2	α_4	0.01
α_2	1	β	0.2
α_3	0.5	ω	0.5

Table 2: Parameters of SWSS in experiments.

The evaluation set of WMT15 has 4 language pairs and each has 500 sentence pairs. WMT16 dataset has 6 language pairs and WMT17 dataset has 7 language pairs, and each has 560 sentence pairs. Performance of a metric is evaluated by Pearson correlation between scores provided by the metric and the human evaluation scores.

4.2 Settings

The parameters of SWSS are tuned on the dataset from WMT17 metric task and are listed in Table 2. We use SpaCy library[1] for word tokenization. Word stems are extracted with Porter stemming algorithm (Porter et al., 1980). UCCA representations are parsed with the pre-trained model of TUPA (Hershcovich et al., 2017).

4.3 Results

SWSS is combined with base models including BLEU, Meteor and Meteor++. Table 1 shows that the combined models lead to significant improvement of Pearson correlation compared to the base models. It can be inferred that adding SWSS as a component to MT evaluation metrics based on lexical similarity can improve their performance. The results also indicates that SWSS performs better than Meteor++, as SWSS regards all semantic core words as copy words while Meteor++ uses only named entities in its experiments. Semantic core

[1] https://spacy.io/

Method	+UCCA	-repr	-len	None
WMT15				
cs-en	**0.609**	0.599	0.606	0.605
de-en	0.638	**0.641**	0.631	0.620
fi-en	**0.668**	0.662	0.666	0.645
ru-en	**0.634**	0.622	**0.634**	0.628
Average	**0.637**	0.631	0.634	0.624
WMT16				
cs-en	0.646	0.648	0.645	**0.649**
de-en	**0.520**	0.512	0.512	0.503
fi-en	**0.548**	0.541	0.543	0.537
ro-en	**0.633**	0.631	0.627	0.626
ru-en	0.578	**0.581**	0.564	0.574
tr-en	**0.638**	0.632	0.627	0.609
Average	**0.594**	0.591	0.586	0.583

Table 3: Results of ablation experiments. "+UCCA" is the complete SWSS model combined with Meteor, "-repr" means the penalties based on UCCA representation (P_S, P_N, P_E) are removed, "-len" means the length penalty is removed, and "None" contains only Meteor without SWSS.

words is clearly a good and large-scale representation of copy words, according to the results.

We also conduct ablation study to figure out whether the penalties we have introduced are redundant or not. The base model is the combination of SWSS and Meteor. If we remove the representation penalties or the length penalty from the base model, it can be found out from Table 3 that the modified models have lower correlation than the complete model. The result with $p < 0.05$ proves that these penalties have a positive effect on the mechanism.

5 Conclusion

In this paper, we propose Semantically Weighted Sentence Similarity (SWSS), which leverages the power of UCCA to identify semantic core words, and then calculates a similarity score for machine translation evaluation. Inspired by copying mechanism used in sequence generation tasks, we argue that semantic core words, which carry important meaning in the sentence, should exist in all good translations. Additionally, SWSS also uses penalties based on the differences between UCCA structures and sentence lengths, including the concept of Scene in UCCA, in order to make the output scores more accurate. Experimental results show that SWSS can produce a higher correlation in MT evaluation when combined with lexical MT evaluation metrics such as BLEU and Meteor.

References

Omri Abend and Ari Rappoport. 2013. Universal conceptual cognitive annotation (ucca). In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 228–238.

Eleftherios Avramidis, Maja Popovic, David Vilar, and Aljoscha Burchardt. 2011. Evaluate with confidence estimation: Machine ranking of translation outputs using grammatical features. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 65–70. Association for Computational Linguistics.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract meaning representation for sembanking. In *Proceedings of the 7th linguistic annotation workshop and interoperability with discourse*, pages 178–186.

Alexandra Birch, Omri Abend, Ondřej Bojar, and Barry Haddow. 2016. HUME: Human UCCA-based evaluation of machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1264–1274, Austin, Texas. Association for Computational Linguistics.

Ondřej Bojar, Yvette Graham, and Amir Kamran. 2017. Results of the WMT17 metrics shared task. In *Proceedings of the Second Conference on Machine Translation*, pages 489–513, Copenhagen, Denmark. Association for Computational Linguistics.

Ondřej Bojar, Yvette Graham, Amir Kamran, and Miloš Stanojević. 2016. Results of the wmt16 metrics shared task. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 199–231.

Christian Buck. 2012. Black box features for the wmt 2012 quality estimation shared task. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 91–95.

Leshem Choshen and Omri Abend. 2018. Reference-less measure of faithfulness for grammatical error correction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 124–129, New Orleans, Louisiana. Association for Computational Linguistics.

Michael Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the ninth workshop on statistical machine translation*, pages 376–380.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. *arXiv preprint arXiv:1603.06393*.

Yinuo Guo and Junfeng Hu. 2019. Meteor++ 2.0: Adopt syntactic level paraphrase knowledge into machine translation evaluation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 501–506.

Yinuo Guo, Chong Ruan, and Junfeng Hu. 2018. Meteor++: Incorporating copy knowledge into machine translation evaluation. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 740–745.

Daniel Hershcovich, Omri Abend, and Ari Rappoport. 2017. A transition-based directed acyclic graph parser for UCCA. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1127–1138, Vancouver, Canada. Association for Computational Linguistics.

Chi-kiu Lo, Anand Karthik Tumuluru, and Dekai Wu. 2012. Fully automatic semantic mt evaluation. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 243–252. Association for Computational Linguistics.

George A Miller. 1998. *WordNet: An electronic lexical database*. MIT press.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Martin F Porter et al. 1980. An algorithm for suffix stripping. *Program*, 14(3):130–137.

Miloš Stanojević, Amir Kamran, Philipp Koehn, and Ondřej Bojar. 2015. Results of the wmt15 metrics shared task. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 256–273.

Elior Sulem, Omri Abend, and Ari Rappoport. 2018. Semantic structural evaluation for text simplification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 685–696, New Orleans, Louisiana. Association for Computational Linguistics.

Hiroshi Uchida and Meiying Zhu. 2001. The universal networking language beyond machine translation. In *International Symposium on Language in Cyberspace, Seoul*, pages 26–27.

Hui Yu, Xiaofeng Wu, Jun Xie, Wenbin Jiang, Qun Liu, and Shouxun Lin. 2014. Red: A reference dependency based mt evaluation metric. In *Proceedings of COLING 2014, the 25th international conference on computational linguistics: technical papers*, pages 2042–2051.

Filtering Noisy Parallel Corpus using Transformers with Proxy Task Learning

Haluk Acarcicek[1], Talha Colakoglu[1], Pinar Ece Aktan[1], Chongxuan Huang[2], Wei Peng[2] *
[1]Turkey R&D AI Enablement Department, Huawei Technologies
{haluk.acarcicek,talha.colakoglu,ece.aktan.hatipoglu}@huawei.com
[2]Artificial Intelligence Application Research Center, Huawei Technologies
{huang.chongxuan,peng.wei1}@huawei.com

Abstract

This paper illustrates Huawei's submission to the WMT20 low-resource parallel corpus filtering shared task. Our approach focuses on developing a proxy task learner on top of a transformer-based multilingual pre-trained language model to boost the filtering capability for noisy parallel corpora. Such a supervised task also helps us to iterate much more quickly than using an existing neural machine translation system to perform the same task. After performing empirical analyses of the finetuning task, we benchmark our approach by comparing the results with past years' state-of-the-art records. This paper wraps up with a discussion of limitations and future work. The scripts for this study will be made publicly available. [1]

1 Introduction

Crawling web has been regarded as a *de facto* approach to produce bitexts, yet the crawled texts are under-qualified often in some aspects to train a proper machine translation system. Under-qualified bitexts present misalignments, no alignments, wrong language pairs, sentences mostly composed of numbers and mathematical formulas, etc. Parallel corpus filtering in this manner holds a critical research area to improve the performance of machine translation systems. WMT organizes a shared task for parallel corpus filtering since 2018 intending to filter our noisy bitexts to this end. The challenge targets low-resource language pairs since 2019.

Many existing filtering methods require multiple layers of elimination by implementing manually engineered features such as length filtering, language identification, normalizing, etc. These hand-picked features work well for a language pair but don't generalize well to another language pair or domain and often bring algorithmic complexity to the overall system.

The LASER (Artetxe and Schwenk, 2019) model achieved state-of-the-art (SOTA) records at the WMT19 shared task on low-resource parallel corpus filtering (Chaudhary et al., 2019). The sentence representation model implemented in LASER provides a means for measuring the similarity between a source and a target sentence. As stated in the future work at Artetxe and Schwenk (2019), there is still space to improve. Utilizing a self-attention mechanism remains future work as the LASER was not built upon the latest transformer architecture (Vaswani et al., 2017). We are also interested in designing a filtering tool that can be efficiently applied to a wide range of language pairs. Pre-trained multilingual language models, such as BERT (Devlin et al., 2018) and RoBERTa (Liu et al., 2019), are exploited to this end.

We make two contributions to the field in this manner. The first contribution is a proposal of approaching the filtering problem as a discrimination task that can be trained with a proxy task and synthetic training data generation (see in Section 3.1). The other contribution is the empirical knowledge learned from an analysis of the finetuning pre-trained multilingual language models on cross-lingual discrimination tasks.

2 Related Work

In the WMT18 shared task, participants mostly used similar techniques in components as prefiltering, scoring the sentence pairs, and using a classifier for feature functions. Teams applied prefiltering rules to eliminate noisy data, including:

- short or lengthy sentences;
- sentence pairs with few words and unbalanced token lengths;

*Corresponding author
[1]https://github.com/WPti/proxy-filter

Proceedings of the 5th Conference on Machine Translation (WMT), pages 940–946
Online, November 19–20, 2020.

- sentence pairs with unmatched names, numbers, web addresses, etc.;

- sentences where a language identifier fails to identify a source or target language type.

Scoring functions were mostly used to correlate qualified texts. Participants also used sentence embeddings (Bouamor and Sajjad, 2018; Axelrod et al., 2011; Artetxe and Schwenk, 2019) altogether with a similarity function to detect the similarity of pairs. The WMT19 shared task focused on low-resource languages, namely Nepali-English and Sinhala-English. Participants mostly applied basic filtering techniques similar to those used in 2018. Chaudhary et al. (2019) used sentence embeddings that were trained on parallel sentence pairs. Another approach was to train a machine translation system on the clean data and then used it to translate the non-English side to make a comparison. Several metrics were used to match sentence pairs such as METEOR, Levenshtein distance, and BLEU.

We found that our work relates to the submission from Bernier-Colborne and Lo (2019). However, their submission was unable to show the effectiveness of the proposed method due to potential issues in the pretraining process. Besides the parallel corpus filtering task, we come across several works utilizing a similar approach. In Yang et al. (2019), BERT rescoring method is more effective at bitext mining than heuristic scoring methods, i.e., marginal cosine distance. In Grégoire and Langlais (2018), a similar negative random sampling technique has been used for generating synthetic bad pairs. Also, attempts to create harder negative pairs were proven effective in bitext mining (Guo et al., 2018).

3 Methodology

Transformer models are currently state-of-the-art systems on most NLP classification and regression tasks. With the emergence of multilingual pretrained models, their cross-lingual capabilities can be exploited with little effort.

3.1 Proxy Task

To treat this problem as a supervised one, we design a proxy learner to model this task. The correctly aligned pairs can be regarded as positive samples in a simple sense for binary classification.

Most of the noise in the corpus originate from ill-aligned sentence pairs. The intuitive idea is to treat the misalignments as synthetic negative samples for our proxy task learner.

Taking random samples of the target sentences for all source sentences was the easiest way to create negative samples. But this results in an easily-classifiable training data which offers little assistance to the low-resource bitext filtering task. We need to create more valuable training data, which is referred to as harder examples.

3.1.1 Generating Harder Examples

Instead of training transformers with easily-discernible random negative samples, we need to create harder examples to confuse the model to boost its performance on the filtering task. We try the following ways to generate harder examples:

Neighborhood Awareness The neighbor sentences in the corpus have a higher chance of sharing common semantics and topics than those randomly extracted from corpus-wide. Alignment slips are most likely to occur in this context. This concept of neighborhood awareness inspires us to generate harder training data. For every positive pair, we create two negative pairs by pairing adjacent sentences of that target sentence with the source sentence. Incorporating this simple strategy may help to boost filtering performance.

Fuzzy String Matching Sampling Instead of randomly sampling negative examples from bitexts, we develop a new sampling strategy inspired by KNN (the k-nearest neighbors algorithm). To create harder examples for finetuning, we sampled lexically similar but semantically different sentences using a fuzzy string search method. [2] For each one of the source sentences (S), we perform a fuzzy search and identify the N similar sentence respecting to the fuzzy string score (F). We set a limit (L) on the F and ignore sentences with similarities over this limit (L) to avoid duplicated or highly related candidates. Then we pair the corresponded target sentences of those N similar sentences with the source sentences to create N negative pairs. We apply a setting with an L value of 60 (in a 100 scale) and N values of 2 and 3 to generate the validation and training data.

Model Architecture	Siamese	Finetuning
Bert-base-Multi-cased	0.62	0.69
Xlm-Roberta-Base	0.84	0.86
Xlm-Roberta-Large	0.88	0.92

Table 1: Model performances on proxy task as accuracy in F1 scores.

3.1.2 Architecture

We explore two candidate architecture in this study, one of which is a Siamese network (Reimers and Gurevych, 2019). The other model is a pre-trained transformer with a binary classification learner to differentiate ok-aligned sentence pairs with their negative counterparts. A comparison between the performance of architecture can be seen in the Table 1.

Sentence Transformers Reimers and Gurevych (2019) adopt a Siamese architecture, which allows us to feed sentence pairs separately to a transformer network like BERT. Each sentence pairs are encoded into fixed-size embeddings connected to a classifier network. Embeddings can be compared using a cosine similarity function at the inference stage. We reach on par performance to the LASER in the WMT19 parallel corpus filtering task (Table 3).

Transformer Finetuning with Pair Classification BERT is a language model introduced by Devlin et al. (2018). A pre-trained BERT model can be finetuned by adding an extra output layer to address many NLP tasks. One of BERT's derivatives is RoBERTa (Liu et al., 2019), and it is essentially very similar to its successor in structure. The authors of RoBERTa discarded the next sentence prediction (NSP) task and altered the mask language modeling task.

We compare multilingual variations of BERT and RoBERTa, which contains both Khmer (km) and Pashto (ps) monolingual data in the pretraining. The multilingual version of the RoBERTa, aka XLM-R (Conneau et al., 2019), performs far superior as it leverages more data in training (Table 1).

3.1.3 Amount of Parallel Data

To observe the effect of the amount of the available parallel corpus on this proxy learner's performance, we try two different data regimes. The orange line in Figure 1 represents a very low resource setting, and we subsampled 2k parallel pairs to mimic that. The blue line represents a 10k subsampled version of the training data. As can be seen from Figure 1, the more we increase the number of parallel sentences used in training the proxy task, the more performance we observe for the proxy task. Other than that, a system using almost as little as 2k parallel sentence pair is enough to beat the benchmark results. The proposed approach is promising for other low-resource domains and applications.

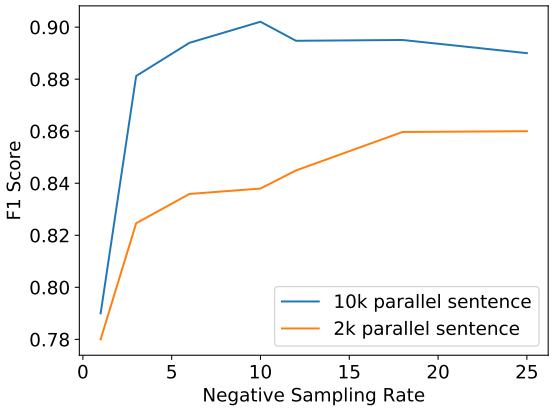


Figure 1: Proxy task validation performance in the face of changing volumes of training data (Pashto - English).

3.1.4 Negative Sampling Ratio

The amount of negative data that can be used in training is analyzed in the prior works (Section 2). Into our observations from Figure 1 and Figure 2, using larger negative ratios leads to better performances. However, it is better to keep the positive/negative ratio to $1:10$ for our datasets with a presence of more parallel data.

We oversample the positive pairs in the finetuning step to balance the positive-negative ratio. But it didn't make a noticeable change in proxy task performance or filtering performance. The immunity of the pre-trained transformer models to the class-imbalance up to 20x is very surprising.

3.1.5 Learning Rate

To prevent the catastrophic forgetting problem in the transformers, we apply a very small ($2e^{-6}$) learning rate with the inverse root scheduler and a warmup step of $1,000$. We also try other learning rate schedulers like cyclic learning rate scheduler (CLR) from (Lee et al., 2020) but couldn't observe any benefit for this task. We suspect CLR may not

[2]https://github.com/seatgeek/fuzzywuzzy

apply to a finetuning process with a small epoch number (i.e., 2 epochs in this study).

3.1.6 Finetuning and Scoring

We add a classification layer on top of XLM-R having $2,048$ hidden units with RELU activations and dropout. On single Nvidia V100 GPU, we finetune our models for 2 epochs without any early stopping. It takes about 6 hours to finetune on the generated datasets. The scoring step is just getting the probability of that pair being positive. Scoring a sentence pair takes $5ms$ on average.

3.2 Rescoring

Bidirectional Scoring Similar to the bidirectional scoring in Chaudhary et al. (2019), we reverse source and target sentences and train two different networks, which produce two different scores (SRC-TRG and TRG-SRC) for a pair. We then combine these two scores under (min, mean, max) strategies. In the "min" strategy, we aim to filter false-positive pairs by keeping the lowest score from the (SRC-TRG and TRG-SRC) for each pair. In "max" strategy, we use the highest score for each pair. And in the "mean" strategy, an average of the scores are applied. We observe that filtering on the "max" score can turn some of the false-negative sentences into true-positives, which increases NMT performance (Table 2).

Strategy	BLEU
SRC-TRG	12.97
TRG-SRC	12.65
Mean	12.42
Min	12.93
Max	13.17

Table 2: NMT results of systems trained after filtering based on different bidirectional scoring strategies (Pashto - English)

Ensembling We ensemble our top 3 trained transformer models under (min, max, mean) strategies and observe a minor improvement on the Pashto-English (ps-en) dataset. On the Khmer-English (km-en) dataset, there is no improvement (Table 3).

3.3 Heuristic Filters

Heuristic filters like overlap filters, length ratio, min-max length, and language identification are applied. For the Pashto-English setup, this step is not beneficial to the overall performance. For the Khmer-English setup, we observe a minor gain (Table 3). It appears that our scoring method can learn heuristic filtering on the fly without reliant on hard-coded heuristic filters.

4 Results

There is a relationship between F1 scores of the proxy task and the final NMT system performance (Figures 1-2). Improvements of the final NMT in the proxy task peaks along with the negative sampling rate and decreases potentially due to overfitting. By looking at the same ratio presented in Figures 1-2, we can conclude a correlation between the performance of the proxy task and that for the filtering task, showcasing the proposed approach's effectiveness.

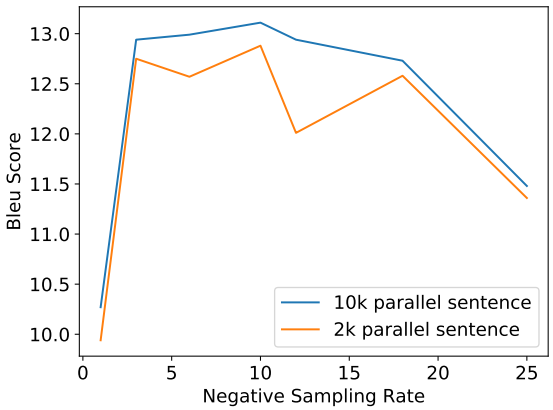


Figure 2: MBART performance of the filtering model (Pashto - English).

WMT20 Here we have presented our NMT performances of the submitted filtering systems in Table 3. Note that we measure all of the development cycles and improvements with the MBART finetuning (Liu et al., 2020). We do not replicate every experiment with training from scratch regime due to resource constraints. As shown in Table 3, our method outperforms the LASER baseline without needing any prefiltering rules and costly marginal KNN scoring method in solving the hubness problem for both language settings.

4.1 Older Tasks

To find how our method generalizes across different filtering scenarios, we test it for the past generations of this shared task.

WMT18 We use the same neural machine translation system defined by the organizers. Our NMT

Method	Pashto-English		Khmer-English	
	Scratch	MBART*	Scratch	MBART*
Baseline(LASER)	9.6	12.2	7.1	10.4
Sentence Transformers	9.7	12.5	7.5	10.6
XML-R finetuning	10.1	12.6	7.7	10.8
+Neighbourhood Awareness	-	12.9	-	-
+Fuzzy String Matching	-	13.0	-	-
+Bidirectional Scoring	-	13.2	-	11.5
+Ensemble Scoring	**10.9**	**13.3**	-	11.5
+Heuristic Filters (3.3)	10.7	13.2	**8.7**	**11.7**

Table 3: NMT scores (BLEU) of the models that trained on a corpus filtered by the specified methods on WMT20 test sets. The bold fonts indicate the SOTA results. * indicates finetuning of the pretrained MBART model which is provided by the organizers.

model using the submission by Junczys-Dowmunt (2018) couldn't reach the reported scores (can be observed in Table 4 for the $10M$ subsampled set). Although our method couldn't match the SOTA results under these settings, it achieves a reasonable score. Note that we only used 10% of the available clean parallel data to accomplish this result. Also, instead of finetuning a multilingual pre-trained model, bilingual models can be tried to avoid the curse of multilinguality (Conneau et al., 2019).

WMT19 Our NMT model using the submission by Chaudhary et al. (2019) couldn't reach the reported scores, as shown in Table 4 for the Nepalese-English (ne-en) set. The mismatches mentioned above with WMT18 and WMT19 are possibly due to a result of using multiple GPUs with distributed optimizers like stated in Koehn et al. (2019). In the low-resource setting, our method can surpass the SOTA results (Table 4).

Task	SOTA	OURS
WMT18 (de-en)	***27.9 (28.62)**	27.53
WMT19 (ne-en)	*6.9 (7.1)	**7.5**

Table 4: WMT18 and WMT19 filtering tasks test results. Note that numbers with "*" represent the submitted score performance under our NMT setup. Those in parenthesis are the reported scores.

5 Conclusions and Future Work

We illustrate our submission to the WMT20 low-resource parallel corpus filtering task. By developing a proxy task learner on top of a transform-based pre-trained language model XLM-R, We are able to improve the filtering capability for noisy data, achieving SOTA results.

The parallel corpus filtering task is recall-oriented. Therefore our model may not be suitable for high-precision jobs. The model has limitations in dealing with short sentences. It can be improved by finetuning on dictionaries or phase-based bitexts. The model performs better in low-resource and high-recall settings.

In our experiments depicted in the subsection 3.1.6, we observe low performances several times. It may appear the model is suffering from the random seeds caused fragility mentioned in Risch and Krestel (2020). A close look ascribes these abnormal results to the randomness in the sampling strategy. We leave this issue to future work.

Different kinds of synthetic noise generation techniques can be adapted to increase the robustness and accuracy of the model. For example in the filtered data we observed several false-positive cases which contains mis-translated numbers:

en reference:

"3) Sonar coverage: 45K at 200KHz"

ps to en translation:

"4) Sonar coverage: 90 at 125KHz"

Training an NMT model on this type of data hurts the translation performance. But this kind of noise can be fixed by altering the numerical values in the clean training data to sample negative pairs for our proxy task. Moreover, all the other synthetically generatable errors like a typo error, one to many alignment errors, etc. can be incorporated into the training data. But its not viable to model those kinds of errors independent from

the language or domain with the naive assumptions and inventing heuristic rules. We believe further researches should focus on domain invariant noise generation techniques.

Acknowledgments

We would like to show our gratitude to colleagues from HTRDC AIE and AARC, Huawei for their support during this work.

References

Mikel Artetxe and Holger Schwenk. 2019. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics*, 7:597–610.

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. 2011. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 355–362, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Gabriel Bernier-Colborne and Chi-kiu Lo. 2019. NRC parallel corpus filtering system for WMT 2019. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 252–260, Florence, Italy. Association for Computational Linguistics.

Houda Bouamor and Hassan Sajjad. 2018. H2@ bucc18: Parallel sentence extraction from comparable corpora using multilingual sentence embeddings. In *Proc. Workshop on Building and Using Comparable Corpora*.

Vishrav Chaudhary, Yuqing Tang, Francisco Guzmán, Holger Schwenk, and Philipp Koehn. 2019. Low-resource corpus filtering using multilingual sentence embeddings. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 261–266, Florence, Italy. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Francis Grégoire and Philippe Langlais. 2018. Extracting parallel sentences with bidirectional recurrent neural networks to improve machine translation. *CoRR*, abs/1806.05559.

Mandy Guo, Qinlan Shen, Yinfei Yang, Heming Ge, Daniel Cer, Gustavo Hernandez Abrego, Keith Stevens, Noah Constant, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Effective parallel corpus mining using bilingual sentence embeddings. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 165–176, Belgium, Brussels. Association for Computational Linguistics.

Marcin Junczys-Dowmunt. 2018. Dual conditional cross-entropy filtering of noisy parallel corpora. In *Proceedings of the Third Conference on Machine Translation*, pages 901–908, Belgium, Brussels. Association for Computational Linguistics.

Philipp Koehn, Francisco Guzmán, Vishrav Chaudhary, and Juan Pino. 2019. Findings of the WMT 2019 shared task on parallel corpus filtering for low-resource conditions. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 54–72, Florence, Italy. Association for Computational Linguistics.

Cedric K. M. Lee, Jianfeng Liu, and Wei Peng. 2020. Applying cyclical learning rate to neural machine translation. *ArXiv*, abs/2004.02401.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *arXiv preprint arXiv:2001.08210*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Julian Risch and Ralf Krestel. 2020. Bagging BERT models for robust aggression identification. In *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*, pages 55–61, Marseille, France. European Language Resources Association (ELRA).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Yinfei Yang, Gustavo Hernández Ábrego, Steve Yuan, Mandy Guo, Qinlan Shen, Daniel Cer, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2019. Improving multilingual sentence embedding using bi-directional dual encoder with additive margin softmax. *CoRR*, abs/1902.08564.

Score Combination for Improved Parallel Corpus Filtering for Low Resource Conditions

Muhammad ElNokrashy [1], **Amr Hendy**[1], **Mohamed Abdelghaffar**[1], **Mohamed Afify**[1], **Ahmed Tawfik**[1] **and Hany Hassan Awadalla**[2]
[1] Microsoft Egypt Development Center, Cairo, Egypt
[2] Microsoft Corporation, Redmond, WA, USA
{muhammad.elnokrashy,a-amhend,mohamed.abdelghaar,mafify,atawfik,hanyh}@microsoft.com

Abstract

This paper presents the description of our submission to WMT20 sentence filtering task. We combine scores from custom LASER built for each source language, a classifier built to distinguish positive and negative pairs and the original scores provided with the task. For the mBART setup, provided by the organizers, our method shows 7% and 5% relative improvement, over the baseline, in sacreBLEU score on the test set for Pashto and Khmer respectively.

1 Introduction

Neural machine translation (NMT) brings significant gains to the field of machine translation. However, it is known to be very sensitive to the quality of parallel data (Khayrallah and Koehn, 2018). This becomes a serious problem when using large but very noisy corpora for training. There is a lot of work on filtering noisy parallel data. For example, the work in (Junczys-Dowmunt, 2018) provides excellent results for large corpora. Low resource languages are even more challenging and the results in (Chaudhary et al., 2019) using multilingual embeddings are very encouraging.

This paper describes the system submitted to the WMT20 Shared Task on Parallel Corpus Filtering for Low Resource Conditions. Due to time limitation our submission covers only sentence pair filtering. However, some of the proposed techniques could be used for sentence alignment and filtering. The task focuses on the Pashto-English and Khmer-English language pairs. It is required that the participants calculate scores to sort very noisy parallel sentence pairs provided for each language. The top scoring pairs leading to 5M tokens on the English side are then used to train machine translation for each language pair. The organizers also provide LASER-based scores as a baseline according to the method in (Chaudhary et al., 2019), with possibly some modifications. We describe our general system architecture followed by development experiments to evaluate the merit of different methods and finally report the performance of our best setup per language, using the fairseq recipe provided for the task for both full training (from scratch), and finetuning (mBART) settings. mBART (Liu et al., 2020) is a recently proposed pretraining method. Models initialized using mBART for both language pairs are provided with the task.

Our proposed approach combine the scores provided by the organizers with the following two scores:

- Margin distance calculated based on custom LASER built for each language using parallel data and a large amount of forward translated mono data provided by the organizers.
- Cosine distance between the embeddings generated by a classifier taking the pretrained LASER as input and trained to distinguish parallel and non-parallel sentence pairs using parallel data provided in the task.

More details on the approach are provided in Section 2. For the mBART setup our method shows 7% and 5% relative improvement in sacreBLEU score on the test set for Pashto and Khmer respectively.

2 System Architecture

In this section we describe the overall system architecture. We start by presenting language-based preprocessing in Section 2.1.The scores provided by the organizers use pretrained LASER (Artetxe and Schwenk, 2019b) and margin distance (Artetxe and Schwenk, 2019a; Johnson et al., 2017). Actually their method obtained state-of-the-art results in WMT19 low resource filtering task for Sinhala,

Proceedings of the 5th Conference on Machine Translation (WMT), pages 947–951
Online, November 19–20, 2020.

Nepali and Hindi and hence is a very strong baseline. Therefore, we opt to construct sentence embeddings using two different methods, namely custom LASER and classifier-based embeddings, to complement the baseline pretrained LASER. Each embedding method is then used to generate a score for each sentence pair and all the scores are combined to form the final score. In the rest of this section, we will describe custom LASER in Section 2.2 followed by classifier-based embeddings in Section 2.3 and finally we outline score combination in Section 2.4.

2.1 Preprocessing

We first preprocess the data using language identification on the source side. The results reported in this paper use the BLING (BLING, 2020) tool from Microsoft but preliminary experiments indicate similar performance using python langid or fasttext (Bojanowski et al., 2017). One interesting feature of BLING is that it returns the percentages of different language constituents of a sentence. In the current implementation we keep sentences that have a language identification score of the source language of 80% or higher. All sentences that do not satisfy the language identification threshold are assigned a very low score and hence are not selected in the final candidate pairs. We will report results without using language identification in the experiments. We generally found it helps a lot for Khmer and actually hurts a bit for Pashto. We expect this is due to the larger Khmer size. Table 1 shows the original number of sentences and those filtered at 80% threshold for Pashto and Khmer.

Language	Original	Filtered (kept)
Ps	1,022,883	615,451 (60.17%)
Km	4,169,574	2,714,664 (65.11%)

Table 1: Number of sentences before and after language filtering

2.2 Custom LASER

Pretrained LASER (Artetxe and Schwenk, 2019b) has 93 languages. Some of these languages are under-represented and others, like Pashto, are completely missing. While similar languages tend to help each other it is clearly beneficial to have a custom LASER trained for the languages of interest. In WMT19 results (Chaudhary et al., 2019), custom LASER trained on a combination of Hindi, Sinhala and Nepali outperformed the pretrained LASER for the filtering task. Here, we build two separate models for Pashto and Khmer. Since both languages have very different origins we thought it is not beneficial to build a combined model but we haven't verified this experimentally. We use the LASERtrain package (Esplà and Sánchez-Martínez, 2019) to train the custom LASER. This package follows the LASER training as given in (Artetxe and Schwenk, 2019b) and provides experiments on BUCC'18 with good results. It is also possible to fine-tune the pretrained LASER using the languages of interest. We will explore this in future work.

Participants in WMT20 are limited to data provided by the organizers. The supplied parallel data for both languages is of rather limited size and is dominated by domain specific data as software localization and religious text and hence we use the provided monolingual text to augment the training data. For each language this is done as follows. We start with the provided sentence scores and filter 5M English tokens as suggested in the task. We use the resulting parallel data to train an mBART initialized MT system. The sacreBLEU scores on the development test set from the organizers and our internal run are shown in Table 2.

Language-pair	Organizers	Internal
Ps-En	12.2	11.6
Km-En	10.6	10.4

Table 2: SacreBLEU for mBART on development test set as provided by the organizers and for internal run.

In addition to the noisy parallel sentences, the organizers provide additional parallel data and monolingual data for both languages. The parallel data comes mainly from OPUS and consists of 290K and 123K pairs for Khmer and Pashto respectively. The monolingual data for Khmer has around 13M sentences while that of Pashto has around 6M sentences. For more information about the sources of these data we refer the reader to (EMNLP, 2020). The resulting internal system is used to forward translate various amounts of monolingual data from the source language into English. We found in preliminary experiments that using around 3M monolingual sentences gives good performance (more

on the evaluation below). These sentences are randomly selected from the monoingual data. The synthetic pairs for Pashto-English and Khmer-English are used to augment the provided clean parallel data to train the custom LASER for each language. The reason we build unidirectional custom LASER is that most of our data is synthetic and the performance of the translation in the opposite direction English-Pashto(Khmer) is expected to be quite poor. In addition, English is very well represented in the pretrained LASER.

It is good to have a way to evaluate the quality of the built custom LASER. Building machine translation (MT) every time is very costly. To this end, we use a BUCC-like setup to evaluate the quality of the custom embeddings. We use the development set for each language pair. For each sentence on the source side we use the corresponding embedding to find the nearest neighbor, based on cosine distance, on the target side and calculate the top-1 accuracy. We do the same in the other direction (target to source) and average both numbers.

$$\text{BUCC}(S) = \frac{1}{2|S|}\left(\sum_i \mathbb{I}\left(\arg\max_j S_{i,j} = i\right) + \sum_j \mathbb{I}\left(\arg\max_i S_{i,j} = j\right)\right) \quad (1)$$

where S is the matrix of pairwise similarity scores for pairs in the source-target development set. Accuracies using pretrained and custom LASER for Pashto and Khmer are shown in Table 3.

Language-pair	Pretrained	Custom
Ps-En	9.56%	31.97%
Km-En	1.04%	39.50%

Table 3: BUCC-like accuracy scores on devtest set of the filtration task

Once the custom LASER of a language is trained it is used to calculate the score of a sentence pair using the margin distance as shown in Equation 2. The margin is implemented efficiently using (Johnson et al., 2017).

$$\begin{aligned}\text{score}(x, y) = &\text{margin}(\cos(x, y), \\ &0.5\,(\text{mean}\,\{\cos(z, x) \mid z \in \text{NN}_k(x)\} \\ &+ \text{mean}\,\{\cos(z, y) \mid z \in \text{NN}_k(y)\}))\end{aligned} \quad (2)$$

2.3 Classifier-Based Scores

In addition to custom LASER presented in the previous section we use scores provided from a classifier trained to distinguish parallel and non-parallel sentence pairs. It takes pretrained LASER embeddings of a sentence pair u and v and transforms them using a fully-connected layer with ReLU nonlinearity. Similar to (Reimers and Gurevych, 2019) it inputs the concatenation $[u_{tr}; v_{tr}; |u_{tr} - v_{tr}|]$, where u_{tr} and v_{tr} are the outputs of the fully connected layer, to a softmax classifier with two outputs representing the positive and negative pairs. The network is trained using the cross-entropy criterion. During testing, LASER embeddings of a sentence pair are passed through the fully connected layer and their cosine distance is calculated as the required score. The rationale is that the transformed embeddings provide better representation to separate positive and negative pairs compared to pretrained LASER.

For each language, the classifier is trained on the positive pairs provided by the organizers. Following (Zhang et al., 2020) for each sentence the negative pair is selected at random from the following:

- Select a sentence from its adjacent sentences within a window size of k (where k = 2 in our experiments).
- Truncate 30-70% words of the sentence.
- Swap the order of 30-70% of the words of the sentence.

After forming the positive and negative data around 500 example pairs, per language, are kept as validation set. The classification accuracy, on the validation set, for Pashto is 97% while that of Khmer is 98.5%.

2.4 Score Combination

Based on the previous sections each input sentence pair x, y has three scores. Assume the pretrained LASER embeddings are x_p and y_p and the custom

LASER embeddings are x_c and y_c. We can write the combined score $S(x, y)$ as follows:

$$S(x, y) = S_{mg}(x_p, y_p) + S_{mg}(x_c, y_c) + S_{cl}(x_p, y_p) \quad (3)$$

where $S_{mg}()$ indicates margin distance and $S_{cl}()$ indicates classifier distance. We choose to use a simple sum instead of using trainable weights because the provided parallel data that could be used to train the weights is very specific and could result in biased estimates of the weights.

We also experiment with minimum-maximum normalization that we found very useful in the case of Khmer. For this normalization each component score in Equation 3 is modified as follows:

$$S_{norm} = \frac{S - S_{min}}{S_{max} - S_{min}} \quad (4)$$

where S_{min} and S_{max} are the minimum and maximum scores over all the pairs.

3 Experimental Results

In this section we first present the results of various experiments to arrive at the final system architecture for both languages in Section 3.1. An internal system with a small architecture is used for fast turn-around. This is followed by running experiments with the final architecture using the official scripts provided by the organizers for both the from scratch and mBART settings.

3.1 Development Experiments

This section outlines various development experiments for Pashto and Khmer. As mentioned above an internal system with a small configuration is used to compare different configurations. The sacreBLEU scores for Pashto are shown in Table 4 while those for Khmer are in Table 5.

	Dev. Set	Test Set
B	7.4	8.4
C	9.3	9.3
B + C	9.2	10.4
B + C + Cl	9.5	10.5
B + C + Cl (BL)	8.3	9.6

Table 4: Pashto Development results (in SacreBLEU) for different configurations on development and test sets. B stands for baseline, C for custom, Cl for classifier and BL for BLING

	Dev. Set	Test Set
B	8.8	7.0
C	4.3	3.6
C (BL)	6.5	5.2
B + C (BL)	9.5	7.6
B + C + Cl (BL)	9.9	8

Table 5: Khmer Development results (in SacreBLEU) for different configurations on development and test sets.B stands for baseline, C for custom, Cl for classifier and BL for BLING.B+C+Cl result for Khmer uses minimum-maximum normalization.

From the two tables we can see that there is some significant difference in behavior between Pashto and Khmer. This can be summarized as follows:

- While custom LASER is better than pretrained LASER for Pashto it is worse for Khmer. We attribute this to the existence of Khmer and absence of Pashto in pretrained LASER. For Pashto, even with some small parallel data and synthetic data we can see some nice gains.
- BLING language filtering is crucial for Khmer while it hurts a bit for Pashto. We attribute this to the larger size and the noisier nature (from the view point of having more English words in the source side) of the Khmer data.
- Even if the custom LASER for Khmer is significantly worse than the pretrained one it helps when combined with the baseline.

3.2 Final Experiments

Based on the above observations, we decided to have our configurations for the final experiments of the two languages as follows:

- Use BLING filtering for Khmer but not for Pashto.
- Use the combined scores of pretrained LASER, custom LASER and classifier for both Pashto and Khmer.
- Experiment with and without min-max normalization.

The results using both from scratch (Full) and mBART (FT) settings as supplied by the organizers are shown in Table 6.

	Mode	Base	Ensemble	
			min-max	No norm
Ps-En	Full	10.04	10.12	10.05
	FT	11.61	11.99	12.38
Km-En	Full	7.16	7.88	6.36
	FT	10.56	11.12	9.02

Table 6: Final results in SacreBLEU on devtest set. Full stands for from Scratch and FT for mBART.

Based on the results in the table our submission used minimum-maximum normalization for Khmer but not for Pashto. By looking into the unnormalized scores we found that for Khmer they tend to be dominated by the classifier score, undermining both the baseline and custom LASER, and hence the normalization helps to bring all scores to the same dynamic range.

4 Summary

This paper present the description of our submission to WMT20 sentence filtering task. By building custom LASER and a classifier to distinguish positive and negative pairs. For the mBART setup our method shows 7% and 5% relative improvement in sacreBLEU score on the test set for Pashto and Khmer respectively.There are a lot of extensions along the proposed directions to improve sentence filtering for the low resource setting.

References

Mikel Artetxe and Holger Schwenk. 2019a. Margin-based Parallel Corpus Mining with Multilingual Sentence Embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3197–3203, Florence, Italy. Association for Computational Linguistics.

Mikel Artetxe and Holger Schwenk. 2019b. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics*, 7:597–610.

BLING. 2020. https://docs.microsoft.com/en-us/azure/cognitive-services/translator/.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Vishrav Chaudhary, Yuqing Tang, Francisco Guzmán, Holger Schwenk, and Philipp Koehn. 2019. Low-resource corpus filtering using multilingual sentence embeddings. *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*.

EMNLP. 2020. Shared Task: Parallel Corpus Filtering and Alignment for Low-Resource Conditions.

Miquel Esplà and Felipe Sánchez-Martínez. 2019. LASERtrain. `https://github.com/transducens/LASERtrain/`.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2017. Billion-scale similarity search with gpus. *arXiv preprint arXiv:1702.08734*.

Marcin Junczys-Dowmunt. 2018. Dual conditional cross-entropy filtering of noisy parallel corpora. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, page 901–908, Belgium, Brussels.

Huda Khayrallah and Philipp Koehn. 2018. On the impact of various types of noise on neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, page 74–83, Melbourne, Australia.

Y. Liu, J. Gu, N. Goyal, X. Li, S. Edunov, M. Ghazvininejad, M. Lewis, and L. Zettlemeyer. 2020. Multilingual denoising pretraining for neural machine translation. *arXiv preprint arXiv:2001.08210v2*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3980–3990, Hong Kong, China. Association for Computational Linguistics.

Boliang Zhang, Ajay Nagesh, and Kevin Knight. 2020. Parallel Corpus Filtering via Pre-trained Language Models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8545–8554, Online. Association for Computational Linguistics.

Bicleaner at WMT 2020: Universitat d'Alacant–Prompsit's submission to the parallel corpus filtering shared task

Miquel Esplà-Gomis[⋆] **Víctor M. Sánchez-Cartagena**[⋆]
Jaume Zaragoza-Bernabeu[†] **Felipe Sánchez-Martínez**[⋆]

[⋆]Dep. de Llenguatges i Sistemes Informàtics, Universitat d'Alacant
E-03690 Sant Vicent del Raspeig (Spain)
{mespla,vmsanchez,fsanchez}@dlsi.ua.es

[†]Prompsit Language Engineering
Av. Universitat s/n. Edifici Quorum III, E-03202 Elx (Spain)
jzaragoza@prompsit.com

Abstract

This paper describes the joint submission of Universitat d'Alacant and Prompsit Language Engineering to the WMT 2020 shared task on parallel corpus filtering. Our submission, based on the free/open-source tool Bicleaner, enhances it with Extremely Randomised Trees and lexical similarity features that account for the frequency of the words in the parallel sentences to determine if two sentences are parallel. To train this classifier we used the clean corpora provided for the task and synthetic noisy parallel sentences. In addition we re-score the output of Bicleaner using character-level language models and n-gram saturation.

1 Introduction

This paper describes the joint submission of Universitat d'Alacant and Prompsit Language Engineering to the parallel corpus filtering shared task at the Fifth Conference on Machine Translation (WMT 2020). Our submission is built upon Bicleaner (Sánchez-Cartagena et al., 2018),[1] a widely-used free/open-source tool for detecting noisy parallel sentences that participated in the 2018 edition of this shared task and ranked fourth out of 17 submissions on one of the sub-tasks. We provide quality scores for the sentence pairs provided by the organiser without re-aligning them.

The 2020 edition of the parallel corpus filtering shared task focuses on two under-resourced Asian languages paired with English: Khmer and Pashto. Khmer (km) is the official language of Cambodia and is spoken circa 16 million people in Cambodia, Vietnam and Thailand.[2] There are about 500k English–Khmer parallel sentences in OPUS,[3] mainly belonging to narrow domains like software products and religion. Pashto (ps) is spoken by around 40 million people in Pakistan and in Afghanistan, where it is official together with Persian.[4] There are around 100k English–Pastho parallel sentence in OPUS, most of which belong to the software domain.

Detecting noisy parallel sentences for under-resourced language pairs, like those addressed in this shared task, is challenging. Pastho is not directly supported by LASER (Schwenk and Douze, 2017), although it supports other Iranian languages, and there are few bilingual resources for building the Bicleaner's models.

Bicleaner is based on a classifier that assesses whether a pair of sentences are mutual translations or not. It is trained on a parallel corpus (positive samples) and on an automatically corrupted version of the same corpus (negative samples). The most important features used by the classifier are lexical similarity scores obtained with the help of probabilistic bilingual dictionaries, which are also extracted from the parallel corpus. Our submission improves the performance of the version of Bicleaner that took part in the 2018 shared task in multiple ways: a new classification algorithm, new lexical features that account for the frequency of the words in the parallel sentences, and a novel way of generating corrupted pairs of sentences. In addition, we re-score the output of Bicleaner combining character-level language models and an n-gram saturation scorer in a linear combination whose parameters are determined by fine-tuning the MBART model provided by the organisers of the shared task.

The rest of the paper is organised as follows. Section 2 describes the Bicleaner classifier whereas Section 3 explains how the score produced by the

[1] https://github.com/bitextor/bicleaner

[2] Wikipedia: https://en.wikipedia.org/wiki/Khmer_language

[3] http://opus.nlpl.eu

[4] Wikipedia: https://en.wikipedia.org/wiki/Pashto

Proceedings of the 5th Conference on Machine Translation (WMT), pages 952–958
Online, November 19–20, 2020.

classifier is combined with the information provided by character-level language models and an n-gram saturation algorithm to produce the submitted score. Section 4 then describes the process followed to build the submission, and Section 5 lists related approaches. The paper ends with some concluding remarks.

2 Bicleaner classifier

Bicleaner is based on an automatic classifier that produces a score for a pair of sentences representing the probability that they are mutual translations. Random Forests (Breiman, 2001), the classification algorithm used in the 2018 submission, has been replaced by Extremely-Randomised Trees (Geurts et al., 2006) because the latter performed best on preliminary experiments.

The Extremely Randomised Trees classification algorithm works by selecting at each internal node the *best* feature from a sub-set of features selected at random from the whole set of features F, and using a random cut-off point. The hyper-parameters controlling the training of these classifiers are therefore the method used to rank the features and select the best one, the size of the subset of features selected at random, and the number of trees to be used. To select the best hyper-parameters we performed a grid search with the following hyper-parameter values. For the ranking we tried with Gini importance (Breiman et al., 1984, Ch. 4) and information gain; for the size of the sub-set of features we tried with $|F|$, $log_2|F|$ and $\sqrt{|F|}$; for the number of trees we tried with 100, 200, 300, 400 and 500.

The features we used can be split in two groups: those that account for the lexical similarity of the two sentences, and those based on shallow properties of the sentences.

2.1 Lexical features

Bilingual lexical similarity is assessed by means of the lexical feature $\mathrm{Qmax}(S, \Theta, d)$, which was first described by Sánchez-Cartagena et al. (2018) and is inspired by the translation probabilities used in statistical machine translation (Koehn, 2009). It is defined as:

$$\mathrm{Qmax}(S, \Theta, d) = \left(\prod_{t \in \Theta} \max_{s \in \mathcal{S} \cup \{\mathrm{NULL}\}} p(t|s; d) \right)^{\frac{1}{|\Theta|}}$$

where, S is a source-language (SL) sentence, $\mathcal{S}$ is a set with the tokens in S, Θ is a set with the tokens in the target-language (TL) sentence T that appear at least once in the SL-to-TL probabilistic bilingual dictionary d, and $p(t|s; d)$ stands for the translation probability of the target token t given the source token s according to the bilingual dictionary d. A smoothing is applied if, for a token t, $\max_{s \in \mathcal{S} \cup \{\mathrm{NULL}\}} p(t|s; d)$ equals zero; in that case, this expression is set to the value of the smallest probability in d divided by 10. One can interpret that, in this case, the dictionary is providing evidence that t is unlikely to be the translation of any of the tokens in S. It is worth noting that this case differs from the case in which a token $t \in T$ does not appear in the dictionary at all; in that case, no evidence, either positive or negative, is available for it. This is why Qmax is only computed for the tokens in Θ instead of doing so for all the tokens in T.

The informativeness of Qmax strongly depends on the coverage of the probabilistic bilingual dictionary used. To measure the coverage of this dictionary, the feature Qmax is complemented with two additional features: $\mathrm{CoverT}(T, d)$, which returns the percentage of unique tokens in T appearing in d, and $\mathrm{CoverTS}(S, T, d)$, which returns the percentage of unique tokens in T that appear in d associated with at least one token in S. All these features are also computed in the reverse direction: $\mathrm{Qmax}(\Theta, S, d')$, $\mathrm{CoverS}(S, d')$, and $\mathrm{CoverST}(T, S, d')$, where d' is a TL-to-SL probabilistic bilingual dictionary.

Even though low-frequency words usually have more discriminatory power (Ramos, 2003), the original formulation of the Bicleaner lexical features did not take into account word frequency in any way. In order to allow the classifier to give different weights to words from different frequency ranks, we re-formulated the lexical features: Qmax now becomes a set of features $\{\mathrm{Qmax}_q(S, \Theta, d, R) \mid q \in [1, 4]\}$. While the summation in the original Qmax was computed for all the tokens in Θ, in Qmax_q it is only computed for those tokens in Θ that appear in the quartile $q \in [1, 4]$ of the ranking of tokens R. R sorts tokens by the logarithm of their relative frequency in a monolingual corpus; in this way, quartile $q = 1$ contains a large amount of tokens with low frequency, while quartile $q = 4$ contains fewer tokens with high frequency.[5] The same adaptation is applied to obtain

[5] Preliminary experiments showed that no gain is obtained by dividing word frequencies in more than four groups.

the set of features $\{\text{CoverS}_q(T, d) \mid q \in [1, 4]\}$ and $\{\text{CoverST}_q(S, T, d) \mid q \in [1, 4]\}$. As in the original Bicleaner, these features were also computed in the reverse direction.

2.2 Shallow features

Shallow features do not make use of bilingual lexical information and are aimed at complementing the lexical features, which may not be reliable enough in sentence pairs with poor dictionary coverage. The shallow features used can be further split into those that model sentence length and those that identify tokens and characters that give hints about the parallelness of a pair of sentences.

Features that model sentence length are based on the assumption that the ratio between the lengths of a pair of parallel sentences is fairly constant for a given language pair. Hence, sentence pairs that deviate too much from this ratio are not likely to be parallel. We measure how close is the ratio of a given pair of sentences to the expected one as the probability mass function of a Poisson distribution. We also provide the raw lengths to the classifier. The complete list of features based on sentence length is the following. Each of these features is computed independently for the SL sentence S and for the TL sentence T of the pair.

- Likelihood of having a TL segment T with length (in tokens) l_T given l_S, the length of the SL segment S, and r_{ts}, the ratio between the length of TL and SL computed on a training parallel corpus; likelihood is computed as $Pr(X = l_T; \lambda = l_S \cdot r_{ts})$. This feature is also computed for S: $Pr(X = l_S; \lambda = l_T \cdot r_{st})$. Note that $Pr(X = k; \lambda = L) = \frac{e^{-L} \cdot L^k}{k!}$.
- Number of tokens in the sentence.
- Number of characters in the sentence.
- Average token length (in characters) in the sentence.

Parallel pairs of sentences are also likely to share numerical expressions, punctuation marks and proper nouns. The following features aim at leveraging that information. Each of these features is computed independently for S and T.

- Number of punctuation marks of each type.
- Proportion of numerical expressions in the sentence that can be found in the other sentence of the pair.
- Proportion of capitalised tokens in the sentence that can be found in the other sentence of the pair.

Finally, character counts can also be considered hints for parallelness. They are taken into account by the following features, which are computed independently for S and T:

- Number of characters in each of the main Unicode classes.
- Number of different characters.
- Number of occurrences of the three most frequent characters, normalised by sentence length.
- Entropy of the string, considering each character as an event whose probability is proportional to the number of occurrences of the character in the sentence.
- Maximum number of consecutive repetitions of the same character.

Overall, 92 shallow features are used.

2.3 Modelling noise

For training the Bicleaner classifier, positive and negative samples are used. The positive samples are those found in the original parallel corpus. The negative samples are generated by corrupting the sentences in that corpus as explained next.

Three types of synthetic noise are applied for corrupting the sentences:

- wrong alignment: parallel segments are randomly re-aligned to produce pairs of segments that are not parallel;
- wrong segmentation: one of the sentences in the pair is truncated: a suffix starting from a random position is removed, therefore emulating an error in sentence segmentation; and
- word replacement: a random number of words in one of the sentences of the pair is replaced by other words with similar frequency as computed on a monolingual corpus.[6]

The amount of corrupted sentences we generated equals the the size of the original parallel corpus,

[6]The ranking of token frequencies R described in Section 2.1 was used for this replacements.

and the three types of synthetic noise were applied in the same proportion. The classifier is therefore trained on a set of sentences twice as large as the original parallel corpus. This strategy differs from the one followed in the 2018 submission (Sánchez-Cartagena et al., 2018) for generating corrupted sentences, where only the "wrong alignment" type of noise was used.

3 Re-scoring

Subsampling 5 million words from the raw corpus based on the score described in the previous section ensures that NMT systems are trained on parallel data. However, some of the selected training parallel samples may not bring useful information and replacing them with other, more informative samples could improve the performance of the resulting NMT systems. We hypothesise that two main reasons could make a pair of sentences which are mutual translations non-informative: i) sentences are not fluent enough and hence very different from those that will be translated with the resulting NMT systems: lists of keywords or website menus are examples of such non-fluent sentences; and ii) the pair of sentences is too similar to other training samples.

To take into account these additional factors, the final score assigned to each sentence pair was computed as follows. First, each sentence received a preliminary score, $\mathrm{prescore}$, computed as:

$$\mathrm{prescore}(s,t) = \lambda \cdot \mathrm{bicleaner}(S,T) + (1-\lambda) \cdot \min(\mathrm{fluency}_s(S), \mathrm{fluency}_t(T)))$$

where S and T are respectively the SL and TL sentence, $\mathrm{bicleaner}$ is the score described in Section 2, and $\mathrm{fluency}_s$ and $\mathrm{fluency}_t$ denote, respectively, fluency scores in the SL and in the TL provided by character language models.[7]

Fluency scores were computed as the normalised perplexity of the sentence according to a 7-gram character language model estimated with KenLM (Heafield, 2011). Normalisation was aimed at placing the perplexities in the $[0,1]$ interval and consisted on a linear transformation that ensured that the values in the raw corpus had a mean of 0.5 and standard deviation of 0.25. Assuming that the perplexities follow a normal distribution, 95% of the values fall into the desired range. Those values with score lower than 0 or higher than 1 after the transformation were set to 0 and 1, respectively.

After computing $\mathrm{prescore}$, sentence pairs were sorted by that score in descending order, and the score of those pairs for which all their 3-grams could be found in sentences with a higher score was multiplied by a penalty β to promote diversity in the subsampled corpus.

The values of the parameters λ and β, that control the contribution of parallelness, fluency and novelty to the final score were optimised so as to maximise the BLEU score obtained after fine tuning the MBART model provided by the task organisers. The Nelder-Mead algorithm (Nelder and Mead, 1965), which does not require gradient computations, was used.

4 Building the submission

This section describes the process followed to build our submission, which comprised selection of training data, corpora preprocessing, classifier training and evaluation of different alternatives for some of the steps.

4.1 Data used

For both language pairs, the classifier training data was built from the concatenation of all the clean parallel corpora provided by the shared-task organisers. The length ratios used in shallow features were computed on the same data, as well as the bilingual dictionaries. In order to build the dictionaries, the parallel sentences were word-aligned with `MGIZA++`.[8] Alignments were symmetrised with the heuristic *grow-diag-final* and the probabilities in the bilingual dictionaries were estimated afterwards by maximum likelihood.

The Wikipedia monolingual corpus provided by the organisers was used to compute the word frequencies for word ranking R as described in Section 2.1. The same monolingual data was used to train character language models. Pashto and Khmer models were trained on the complete data. A different English language model was trained for each language pair on a random sample of the

[7] Values of λ close to 1.0 make lists of keywords or website menus that are mutual translations to have the highest scores. Value of λ around 0.5 make the top scored segment pairs to be fluent, complete grammatical sentences. Values of λ close to 0.0 make fluent but non-parallel sentences to receive the highest scores.

[8] https://github.com/moses-smt/mgiza.git

English Wikipedia corpus that matched the size of the Pashto/Khmer Wikipedia corpus.

4.2 Pre-filtering

The clean parallel data provided by the organisers was filtered before their use. Those parallel sentences in which at least one side contain less than 20% of characters in the Unicode range of the corresponding language were discarded. The remaining parallel sentences were deduplicated.

The raw sentence pairs to be scored were also pre-processed with a series of heuristic rules: the score was set to zero if any of the conditions was met. These rules were aimed at detecting segments with evident flaws and speeding up the subsequent steps. The rules were aimed at detecting the following defects in the parallel sentences:

- Wrong language: same Unicode filtering applied to the clean corpora (see above).
- Too long sentences: those with more than 1024 characters.
- Untranslated: SL and TL segments are identical after removing numerical expressions and punctuation marks.
- Not fluent: the sentence contain elements such as URLs, arithmetic operators, too many parentheses, escaped Unicode characters, and other common defects that arise when crawling parallel corpora from the web. These elements were detected by means of regular expressions.

4.3 Tokenisation and word segmentation

Tokenization and subword segmentation have shown to improve the recall of the probabilistic dictionaries used to obtain the lexical features described in Section 2.1. We experimented with the following tokenisation and subword segmentation methods, which were applied to the clean data as well as to the raw sentences to be scored:

- Rule-based tokenisation (`tok`) for Pashto, Khmer and English, as provided by the tool Polyglot (Al-Rfou et al., 2013);
- Rule-based tokenisation plus word morphological segmentation with Morfessor (`tok-morph`). For this we used, after tokenisation, the pre-trained models for Morfessor (Virpioja et al., 2013) included in Polyglot.

4.4 Training Bicleaner

As previously mentioned, the probabilistic bilingual dictionaries were obtained from the same parallel corpus used to train the classifier. This strategy has an important drawback. While almost all words would be found in the bilingual dictionaries when training the classifier, the coverage would be much smaller when classifying the raw sentences because of the small amount of parallel data available. In order to close the gap between training and classification, we removed some dictionary entries during training. Specifically, we removed the least frequent entries so as to ensure that the coverage of the truncated dictionaries on the training data matches the coverage of the full dictionaries on the raw sentences to be scored.

4.5 Results

Table 1 depicts the results obtained on the development environment during the preparation of the submission. The system that produced the scores for our final submission is shown in bold.

We firstly evaluated the different tokenisation alternatives described in Section 4.3, and applied the re-scoring scheme described in Section 3 on top of the best performing one. The results show that the tokenisation with Polyglot without any kind of subword segmentation (`tok`) leads to the best results. It is also worth mentioning the poor performance obtained with morphological segmentation, which needs to be studied more carefully. Moreover, re-scoring for increased fluency and diversity further improved the results.

Table 1 also shows the results obtained by the baseline LASER model,[9] which was consistently outperformed by Bicleaner. Comparing the results of the version of Bicleaner used in this submission with that used in 2018 also shows that the changes introduced bring a positive impact.

5 Related work

A shared task on parallel corpus filtering was part of the WMT conference programme for the first time in 2018 (Koehn et al., 2018). That year the task was targeted at a high-resource scenario. NMT models, which already provide the probability of a

[9] These results do not exactly match those published at `http://www.statmt.org/wmt20/parallel-corpus-filtering.html`, probably because of differences in the GPU hardware or random initialization seed.

	Khmer–English		Pashto–English	
System	**fairseq**	**MBART**	**fairseq**	**MBART**
LASER (baseline)	6.80	10.33	9.55	11.50
Bicleaner 2018 tok	7.45	10.16	10.11	11.85
Bicleaner 2020 tok	7.76	10.66	10.10	12.35
Bicleaner 2020 tok-morph	7.33	10.56	8.64	10.94
Bicleaner 2020 tok + re-score	8.25	11.18	10.53	12.80

Table 1: BLEU scores obtained by the different configurations evaluated for Khmer–English and Pashto–English on the development environment provided by the organisers.

TL sentence given an SL sentence, emerged as the dominant approach (Junczys-Dowmunt, 2018).

Last year's edition was focused on a low-resource scenario (Koehn et al., 2019), where parallel data big enough to build NMT models that provide reliable TL probability distributions was not available. The best performing model was LASER, a method based on multilingual sentence embeddings (Chaudhary et al., 2019) that takes advantage of the data available for multiple language pairs. In fact, a LASER model trained on 93 languages is the baseline model published by the organisers for this edition of the shared task.

Unlike LASER, our submission is mainly based on lexical similarity scores analogous to those used in statistical machine translation. They are computed only on parallel data, without any kind of transfer learning from other language pairs. The approach we follow to detect sentences that are mutual translations is similar to the one by Munteanu and Marcu (2005) for detecting parallel sentences in comparable corpora. However, we use a larger set of shallow features not related to lexical similarity and follow a more sophisticated method for generating negative samples.

Concerning our re-scoring strategy for including information about fluency and diversity, participants from past editions also used these attributes to score sentences. For instance, Axelrod et al. (2019) and Vázquez et al. (2019) devised a scoring strategy under the assumption that parallel sentences should have similar monolingual language model perplexities, and many other submissions included a penalty for repetitive sentences (González-Rubio, 2019; Erdmann and Gwinnup, 2019; Bernier-Colborne and Lo, 2019). Nevertheless, to the best of our knowledge, our approach is the first one that directly optimises the weight of these attributes towards an automatic translation evaluation metric.

6 Concluding remarks

We described the joint submission of Universitat d'Alacant and Prompsit Language Engineering to the parallel corpus filtering shared task at the Fifth Conference on Machine Translation (WMT 2020). Our submission is based on Bicleaner, an open source tool based on a classifier that uses lexical similarity features inspired in the translation probabilities used in statistical machine translation.

We presented a series of improvements over the version of Bicleaner that participated in the 2018 edition of the shared task, namely a better classifier, more sophisticated generation of negative samples and a reformulation of the lexical similarity scores which takes into account word frequency. We showed that these improvements are effective and they allowed our submission to outperform LASER, a state-of-the-art method based on multilingual sentence embeddings. Moreover, combining Bicleaner scores with scores that account for fluency and diversity further improved the results.

We plan to keep exploring subword segmentation algorithms that help to fight data sparseness when computing lexical similarity scores with the help of bilingual dictionaries. We also aim at integrating word embeddings into lexical similarity scores, which would allow us to leverage monolingual data in a more effective way.

Acknowledgments

Work funded by the European Union through projects GoURMET and ParaCrawl. GoURMET —Global Under-Resourced Media Translation, grant agreement number 825299— is funded through the H2020 research and innovation programme. ParaCrawl —actions numbers 2017-EU-IA-0178 and 2018-EU-IA-0063– is funded under the Automated Translation CEF Telecom instrument managed by INEA at the European Commission.

References

Rami Al-Rfou, Bryan Perozzi, and Steven Skiena. 2013. Polyglot: Distributed word representations for multilingual nlp. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 183–192, Sofia, Bulgaria. Association for Computational Linguistics.

Amittai Axelrod, Anish Kumar, and Steve Sloto. 2019. Dual monolingual cross-entropy delta filtering of noisy parallel data. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 245–251, Florence, Italy. Association for Computational Linguistics.

Gabriel Bernier-Colborne and Chi-kiu Lo. 2019. NRC parallel corpus filtering system for WMT 2019. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 252–260, Florence, Italy. Association for Computational Linguistics.

Leo Breiman. 2001. Random forests. *Machine Learning*, 45(1):5–32.

Leo Breiman, Jerome Friedman, Charles J. Stone, and R.A. Olshen. 1984. *Classification and Regression Trees*. Taylor & Francis.

Vishrav Chaudhary, Yuqing Tang, Francisco Guzmán, Holger Schwenk, and Philipp Koehn. 2019. Low-resource corpus filtering using multilingual sentence embeddings. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 261–266, Florence, Italy. Association for Computational Linguistics.

Grant Erdmann and Jeremy Gwinnup. 2019. Quality and coverage: The AFRL submission to the WMT19 parallel corpus filtering for low-resource conditions task. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 267–270, Florence, Italy. Association for Computational Linguistics.

Pierre Geurts, Damien Ernst, and Louis Wehenkel. 2006. Extremely randomized trees. *Machine Learning*, 63(1):3–42.

Jesús González-Rubio. 2019. Webinterpret submission to the WMT2019 shared task on parallel corpus filtering. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 271–276, Florence, Italy. Association for Computational Linguistics.

Kenneth Heafield. 2011. KenLM: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland. Association for Computational Linguistics.

Marcin Junczys-Dowmunt. 2018. Dual conditional cross-entropy filtering of noisy parallel corpora. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 888–895, Belgium, Brussels. Association for Computational Linguistics.

Philipp Koehn. 2009. *Statistical machine translation*. Cambridge University Press.

Philipp Koehn, Francisco Guzmán, Vishrav Chaudhary, and Juan Pino. 2019. Findings of the WMT 2019 shared task on parallel corpus filtering for low-resource conditions. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 54–72, Florence, Italy. Association for Computational Linguistics.

Philipp Koehn, Huda Khayrallah, Kenneth Heafield, and Mikel L. Forcada. 2018. Findings of the WMT 2018 shared task on parallel corpus filtering. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 726–739, Belgium, Brussels. Association for Computational Linguistics.

Dragos Stefan Munteanu and Daniel Marcu. 2005. Improving machine translation performance by exploiting non-parallel corpora. *Computational Linguistics*, 31(4):477–504.

John A. Nelder and Roger Mead. 1965. A Simplex Method for Function Minimization. *The Computer Journal*, 7(4):308–313.

Juan Ramos. 2003. Using TF-IDF to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, volume 242, pages 133–142. New Jersey, USA.

Víctor M. Sánchez-Cartagena, Marta Bañón, Sergio Ortiz-Rojas, and Gema Ramírez. 2018. Prompsit's submission to WMT 2018 parallel corpus filtering shared task. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 955–962, Belgium, Brussels. Association for Computational Linguistics.

Holger Schwenk and Matthijs Douze. 2017. Learning joint multilingual sentence representations with neural machine translation. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 157–167, Vancouver, Canada. Association for Computational Linguistics.

Raúl Vázquez, Umut Sulubacak, and Jörg Tiedemann. 2019. The University of Helsinki submission to the WMT19 parallel corpus filtering task. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 294–300, Florence, Italy. Association for Computational Linguistics.

Sami Virpioja, Peter Smit, Stig-Arne Grönroos, and Mikko Kurimo. 2013. Morfessor 2.0: Python implementation and extensions for morfessor baseline. Technical report.

An Exploratory Approach to the Corpus Filtering Shared Task WMT20

Ankur Kejriwal
Department of
Computer Science
Johns Hopkins University
akejriw2@jhu.edu

Philipp Koehn
Center for Language and
Speech Processing
Johns Hopkins University
phi@jhu.edu

Abstract

This document describes an exploratory look into the Parallel Corpus Filtering Shared Task in WMT20. We submitted scores for both Pashto-English and Khmer-English systems combining multiple techniques like monolingual language model scores, length based filters, language ID filters with confidence and norm of embedings.

1 Introduction

For this task the participants were provided with a corpus of parallel data in Pashto-English (ps-en) and Khmer-English (km-en). Additional parallel and monolingual datasets were also provided. The task organizers built neural machine translation (NMT) systems from the scores produced, based on parallel training sets of 5 million words. These systems are sensitive towards noise (Khayrallah and Koehn, 2018) and thus, it becomes important to separate the useful data from the noise. We view the task as data that passes through a pipeline of filters in order to give us the best possible selection of 5 million words in the end.

We determined that language ID filtering is a very strong pre-processing filter and inducting confidence scores is not needed. We also determined that monolingual Language models can help us in selecting sentences even if both the source and target language models are independent of each other. Finally, using using the length of a sentence as a filter helps us create a better NMT system.

We also learn that statistical intuitions do not easily extend to neural embeddings.

2 Baseline

The idea behind our baseline system is to use the cosine distance between two multilingual representations as a notion of parallelism between sentences embedded in the same space. The tool we use for this is LASER[1] which uses an encoder-decoder architecture to train a multilingual sentence representation model. It has been shown by Artetxe and Schwenk (2018b) that LASER is effective at zero-shot cross-lingual natural language inference in the XNLI dataset which makes it promising for this task involving low-resource languages. Koehn et al. (2019) has shown this to be a strong baseline.

We follow the work done by Artetxe and Schwenk (2018a) and begin by generating multilingual sentence embeddings using LASER. The LASER score is a function of the margin between the cosine similarity between a given candidate and it's k nearest neighbors[2]:

Let f(x,y) =

$$\sum_{z \epsilon NN_k(x)} \frac{cos(x,z)}{2k} + \sum_{z \epsilon NN_k(y)} \frac{cos(y,z)}{2k}$$

LASER score(x,y) = margin(cos(x,y),f(x,y))

We experiment with the following definitions of margin:

- **Absolute:** margin(a,b) = a
  Essentially just cosine similarity.
- **Distance:** margin(a,b) = a - b
  Subtracting the average cosine similarity from that of the given candidate. We use this when there are certain points that are extraordinarily close to many other data points and thus, degrade the quality of nearest neighbors.
- **Ratio:** margin(a,b) = a / b
  This is the ratio between the candidate and the average cosine of its nearest neighbors in both directions.

[1] https://github.com/facebookresearch/LASER

[2] cos(x,y) here refers to cosine similarity between the vectors x and y

Proceedings of the 5th Conference on Machine Translation (WMT), pages 959–965

We also experiment with the following techniques of candidate generation:

- **Intersection:**
  Each source sentence is aligned with exactly one best scoring target sentence. Some target sentences may be aligned with multiple source sentences or with none. We repeat this process in the opposite direction and take the intersection of the 2 alignments

- **Max score:**
  We repeat the process used to generate candidates in Intersection, except we select the alignment with the highest score instead of discarding all inconsistent alignments

We find that the best settings were margin set to ratio, candidate generation set to max-score and k set to 4. Note that this list of nearest neighbors does not include duplicates, so even if a given sentence has multiple occurrences in the corpus, it would have (at most) one entry in the list (Chaudhary et al., 2019). These scores are in the range of [0,1].

The BLEU scores obtained by these systems are 11.16 for Pashto and 9.65 for Khmer.

3 Language ID

For our first step, we try to predict the most probable language of a given sentence using use fastText (Joulin et al., 2016). We use the pre-trained model released by the authors that is trained to identify over 170 languages including Pashto, Khmer and English. The intuition behind it is that when working in a bilingual setting, sentences from other languages or code-mixed sentences will be harmful to the MT system. We call this simple language ID

if source = ps|km and target = en **then**
  langID score = LASER score
else
  langID score = 0
end if

An additional idea that we incorporate is how confident we are when predicting the language of a sentence. For example, if we have a sentence where the probability of the target sentence being English is 0.9 and we have another sentence where the probability of the target sentence being English is 0.3, then given the same LASER scores, we would want to give preference to the sentence pair where the probability of the target sentence is 0.9. We do this for both the source and target language.

We try to only keep sentence pairs where we are confident about both the source and target language being correct. We implement this notion by setting a cutoff c for the language ID probability. We call this confident language ID

if prob(src=ps|km) $>$ c and prob(tgt=en) $>$ c **then**
  confidence = prob(src=ps|km) $\cdot$ prob(tgt=en)
else
  confidence = 0
end if
langID score = LASER score $\cdot$ confidence

Where prob(p=q) is the probability of sentence p being from language q.

We show our results for Pashto in Table 1 and make the following observations.

- There is an overall increase in BLEU.
- Simple language ID seems to be better than Confident language ID by a small margin.
- Confident language ID, has a local minima at a cutoff of 0.75
- The scores in confident language ID tend to decrease sharply after a cutoff of 0.8.

This leads us to believe that while good for identifying the language, the confidence scores are not as strong as the LASER scores.

We experiment by including scores if they are within the top 3 predicted languages but see no significant change in scores.

We also experiment by adding the confidence instead of multiplying it in confident language ID but see no significant change in the BLEU scores.

As a result of our observations, we only perform Simple Language ID for Khmer, giving us a BLEU score of 11.51 for Pashto and 10.04 for Khmer.

4 Norm of embeddings as a filter[3]

Liu et al. (2020) showed us that the norm of an embedding can represent how frequent and how context insensitive the embedding is. Essentially, smaller norms represent frequent and context-insensitive rare words. There is an implicit assumption here that the embedding size is large enough to incorporate all the information present in a sentence.

[3] *Note: Throughout this section we shall be using the terms "vector" and "embedding" interchangeably.*

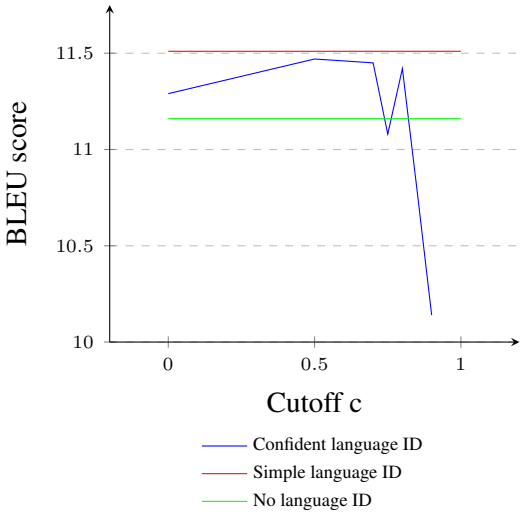


Figure 1: Cutoff vs BLEU scores for Pashto

For a vector $x_i = [x_{i1}, x_{i2}, ..., x_{in}]$ the norm of the vector is

$$norm(x_i) = \sqrt{x_{i1}^2 + x_{i2}^2 + ... + x_{in}^2}$$

We interpret the norm of an embedding as a measure of the confidence we have in the embedding of a sentence. The reason we do this is that neural methods by their very nature are data hungry and susceptible to noise. We are working in a low-resource condition because of which it will be harder to learn about sentences with context-sensitive words. Additionally, it would be harder to learn about sentences with low-frequency words making their embeddings less reliable, thus, leading to a lower quality MT system.

We run 2 sets of experiments on the LASER embeddings of the sentences.

- We assume that the elements in each vector are comparable, i.e. on the same scale. We simply take the norm of the embedding in this case.

- We assume that the elements in a vector are not directly comparable. In this case, we compute the z-score of each element and take the norm of the z-scores. z-score can be thought of as how many standard deviations is a given element away from it's mean. Thus it gives each element a relative value, making them comparable. We finally take the norm of the z-scores.

  $$z\text{-}score = \frac{x-\mu}{\sigma}$$

  where μ is the mean and σ is the standard deviation of that particular element across all the vectors that have a non-zero langID score.

The langID scores we have until now are in the range [0,1], while the norm of the embeddings theoretically have a range of $[0, \infty)$. To ensemble the langID scores with the norm-scores, we need to bring the distributions within a comparable range. We do this by applying min-max normalization.

Let $x = [x_1, x_2, ..., x_n]$
where
x_i = norm (embedding (i^{th} vector))
or
x_i = norm (z-score (embedding (i^{th} vector)))
for all vectors with langID score $\neq 0$ **do**

$$\text{normalized } (x_i) = \frac{x_i - min(x)}{max(x) - min(x)}$$

end for
norm score = langID score - normalized

We observe that in both the cases, there are very few observations with a really high norm because of which over 95% of the norm scores remained the same up to 6^{th} or more decimal place. To counter this, we set the langID scores of these embeddings to 0 and repeat the process recursively till we see an impact on the 5 million token subselected corpus.

We make the following observations

- We see a drastic reduction in the Pashto BLEU scores from 11.51 to 9.76 when we use z-scores.

- When we do not use the z-scores, we see a decrease in the BLEU score from 11.51 to 11.37 for Pashto. and an increase from 10.04 to 10.05 for Khmer. In both the cases the change is really small and not significant.

- Manual observation here shows that sentences with big URL's were filtered out automatically without any explicitly stated rule.

- Being more aggressive with the number of loops led to a drastic decrease in BLEU scores.

We finally ran experiments where we gave a preference to only large norms and an experiment where we gave a preference to both very large norms and to very small norms. In both the cases we had really bad scores leading to the conclusion that the norm of LASER embeddings is not a good

filter. Because of this poor performance when relied on aggressively, we decided not to use it in our final submission.

We also ran the same experiments using fasttext generated embeddings (Bojanowski et al., 2017) but that had poor results as well.

5 Monolingual Language Models

The motivation behind using Monolingual Language Models is that we want to learn about sentence pairs that have a high likelihood of coming up in the test data. Ideally we would want both the sides of the corpus to have a high probability of coming up but we also realize it will often not be the case. Thus, we make these language models independent of each other. We take inspiration from Axelrod et al. (2019) and modify the work of Junczys-Dowmunt (2018) to come up with a language model filter. Junczys-Dowmunt (2018) achieved the highest ranking score in WMT'18 and to do so they define $H_M(.|.)$ as the word-normalized conditional cross-entropy of the probability distribution $P_M(.|.)$ for a model M:

$$H_M(y) = -\frac{1}{y}\sum_{t=1}^{|y|} logP_M(y_t|y_{<t})$$

We use this as a measure of how fluent a given sentence is. Lower scores indicate a better sentence in this case.

While Axelrod et al. (2019) do create n-gram language models, they hope that language models trained on similar but not parallel texts to have similar perplexities over each half of a parallel test set of the parallel corpus. This method does not leverage the simple fact that we have more data for one of the languages. This method also assumes a close relationship between the frequencies of letters which might not always be the case. Finally it does not leverage the expressive power of neural language models. In order to improve on this, we propose Neural Language Models that are completely independent of each other.

5.1 Pre-processing

We first tokenize our data using Sentencepiece (Kudo and Richardson, 2018). We set an upper limit of 5,000 on the vocabulary size for Pashto and Khmer, and an upper limit of 50,000 for the English vocabulary. This is done at both the character and word level and also both with and without splitting at whitespace. We also reverse the data to simulate right-to-left prediction of words. Thus we have 8 possible tokenizations for every possible sentence.

–arch transformer_lm
–dropout 0.1
–optimizer adam
–adam-betas '(0.9, 0.98)'
–weight-decay 0.01
–clip-norm 0.0
–lr 0.0005
–lr-scheduler inverse_sqrt
–warmup-updates 4000
–warmup-init-lr 1e-07 –patience 30

Figure 2: Language Model: Transformer architecture

–arch transformer_lm_wiki103
–max-lr 1.0
–t-mult 2
–lr-scheduler cosine
–lr-shrink 0.75
–warmup-init-lr 1e-07
–min-lr 1e-09
–optimizer nag
–lr 0.0001
–clip-norm 0.1 –patience 30

Figure 3: Language Model: Wiki103 architecture

5.2 Language Model Architecture

For our Language Models, we use fairseq (Ott et al., 2019) to implement the architecture given in Figure 2.

We also create a Language Model using the architecture given by Baevski and Auli (2019) with parameters as given in Figure 3.

We use 2 different models because while it is tempting to use deeper and more sophisticated models, we need to have enough data to train it sufficiently. If sufficient data is absent, it is in general better to train simpler models.

In total we have 16 language models for each language. In each case, we train the model in 2 further ways. In the first case, we keep the CommonCrawl monolingual data as the training set and keep the Wikipedia monolingual data as the development set. In the second case, we augment the CommonCrawl data with Wikipeda data by oversampling. We make the number of lines taken by the wikipedia data be between 40-50% of the number of lines taken by the CommonCrawl data.

During training, the word level language models that were not split on white-space were taking too much time to train. As a result we had to halt their training and use the best checkpoint achieved till then.

5.3 Evaluating the Language models

We evaluated the Language Models by using their word normalized cross entropy as defined in Equation 5. We find that the scores we got were extremely similar whether we used just the CommonCrawl data or whether we augmented it with the Wikipedia data. In addition, the Transformer architecture gave us better results.

We evaluate our created development set and find that the word-level left-to-right and character level left-to-right language models had the lowest perplexities of their respective groups.[4] Additionally, the language models created for Pashto were relatively much stronger than the language models created for Khmer simply because of the script in which Khmer is written.

5.4 Normalising LM-scores

Once again, the values of langID scores range between [0,1] and our language models scores range between [0,∞]. In order to ensemble them, we need to bring them to a comparable scale. We again try to use min-max normalization to change the range of the cross entropy values to be [0,1]. Once again we run into the same problem where the value of maximum is so high that the change bought about in langID scores was negligible. Instead of going for a recursive approach, we take a more aggressive approach this time. We average over the cross entropy values of sentences that we would have selected using langID scores and we replace the max(crossEntropy) with an empirically chosen value close to it.

Let $x = [x_1, x_2, ..., x_n]$
where x_i = cross entropy (i^{th} vector)
for all vectors **do**

$$\text{new entropy } (x_i) = \frac{x_i - min(x)}{max(x) - min(x)}$$

end for
LM score = langID score - new entropy

We display the results for only Pashto and a few English Pashto ensemble models in Figure 4.

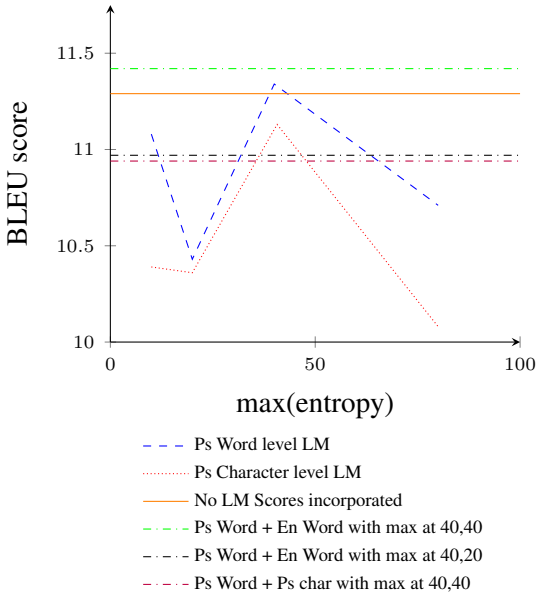


Figure 4: BLEU score vs max(entropy) for Pashto

We can also see that just the word level language models perform better than both the character level models and the Word + Character ensemble models.[5] Similar trends were observed for Khmer.

5.5 Length-based filters

We observed that our language model had a tendency to pick proper nouns. While we want our language model to learn about names, we don't want to select them over sentences because our translation is model is based on sentences. To counteract this, we decided to add a simple length based filters to Pashto and English. Since they both used white spaces and have an almost 1 to 1 mapping, we added a penalty of -1 to their score if any of the sentences were below 0, 5 and 8 in length. We call this cutoff value lc. We decide on a penalty of -1 because the maximum score that any sentence can get right now is 1. This results in that sentence not being selected at all. Following work described in Koehn et al. (2019) we also add a penalty term of -1 whenever either the source or the target sentence was over 3 times it's counterpart in length. We call this length ratio cutoff lr

if len(src) $<$ lc and len(tgt) $<$ lc **then**
    LM score = LM score -1
end if
if len(src)/len(tgt) $>$ lr or len(tgt)/len(src) $>$ lr **then**
    LM score = LM score - 1

[4]Experiments on ensembling these models were still running at the time of submission

[5]We observed that the perplexities on the english side tended to be about half of Pashto and one fourth of Khmer.

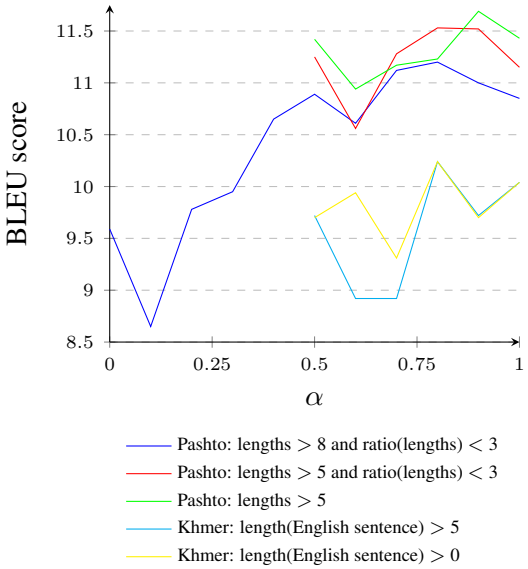


Figure 5: Change in BLEU scores with respect to the amount of weight given to Language Models

end if

In the Khmer-English case, we only experimented with a sentence length on the English side since there were no clear demarcations in Khmer.

if len(tgt) $<$ lc **then**
LM score = LM score -1
end if

We then experiment with a penalty if there is an overlap of more than 60% between the tokenized texts. However this doesn't show any significant changes.

5.6 Assigning Weights to different scoring mechanisms

While we tried to give an equal weight to the Language model and LASER scores, we had no good reason to believe that we should. We introduced a hyperparameter α which lies between 0 and 1, and we change the equation of LM score to be

Let $x = [x_1, x_2, ..., x_n]$
where x_i = cross entropy (i^{th} vector)
for all vectors **do**

$$\text{new entropy } (x_i) = \frac{x_i - min(x)}{40 - min(x)}$$

end for
LM score = (α)(langID score) - (1-α)(LM score)

We replaced the maximum with the value 40 from our findings in Section 5.4

We first run this on Pashto sentences with lc $>$ 8. We use those results to narrow our seaarch with lc $>$ 5 and lc $>$ 0. We then use those results to run experiments for Khmer. The combined results are shown in Figure 5

From figure 5 we can see that we have to local maximas around 0.5 and 0.8 and a global maxima at 0.8. In some cases the global maxima is at 0.9 and in some at 0.8. This leads us to believe that these are the most suitable values for the task.

6 Final Submission

Our final pipeline is as follows:

1. Obtain LASER scores for each sentence pair
2. Pass it through a language id filter where we set the LASER score to 0 if either the source or target language doesn't match
3. score the source and target half of the parallel corpus using monolingual language models
4. combine the language model scores with LASER scores
5. Apply a length based filter to remove sentences that don't provide too much information

We submit what we believe to be the 3 most robust solutions we have for each language pair. For Pashto, we apply the language id filter, then we use the Transformer architecture language model, set lr to 3 and set(α, lc) to (0.8, 5), (0.9, 0) and (0.9, 5). For Khmer, we apply the language id filter, then we use the Transformer architecture language model and set (α, lc) at (0.8, 5) and (0.8, 6). Apart from that, we also submit a scores with a filter checking for token overlap over 30%. The (α, lc) is set to (0.8, 5). At the time of submission our best score for Pashto is 11.69 and the best score for Khmer is 10.24 on the development set. The findings of the shared task presented by Koehn et al. (2020)

References

Mikel Artetxe and Holger Schwenk. 2018a. Margin-based parallel corpus mining with multilingual sentence embeddings.

Mikel Artetxe and Holger Schwenk. 2018b. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond.

Amittai Axelrod, Anish Kumar, and Steve Sloto. 2019. Dual monolingual cross-entropy delta filtering of noisy parallel data. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 247–253, Florence, Italy. Association for Computational Linguistics.

Alexei Baevski and Michael Auli. 2019. Adaptive input representations for neural language modeling. In *International Conference on Learning Representations*.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Vishrav Chaudhary, Yuqing Tang, Francisco GuzmÃ¡n, Holger Schwenk, and Philipp Koehn. 2019. Low-resource corpus filtering using multilingual sentence embeddings. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 263–268, Florence, Italy. Association for Computational Linguistics.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Marcin Junczys-Dowmunt. 2018. Dual conditional cross-entropy filtering of noisy parallel corpora. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 901–908, Belgium, Brussels. Association for Computational Linguistics.

Huda Khayrallah and Philipp Koehn. 2018. On the impact of various types of noise on neural machine translation. *arXiv preprint arXiv:1805.12282*.

Philipp Koehn, Vishrav Chaudhary, Ahmed El-Kishky, Naman Goyal, Peng-Jen Chen, and Francisco Guzmán. 2020. Findings of the WMT 2020 shared task on parallel corpus filtering and alignment. In *Proceedings of the Fifth Conference on Machine Translation: Shared Task Papers*.

Philipp Koehn, Francisco GuzmÃ¡n, Vishrav Chaudhary, and Juan Pino. 2019. Findings of the wmt 2019 shared task on parallel corpus filtering for low-resource conditions. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 56–74, Florence, Italy. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Xuebo Liu, Houtim Lai, Derek F Wong, and Lidia S Chao. 2020. Norm-based curriculum learning for neural machine translation. *arXiv preprint arXiv:2006.02014*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Dual Conditional Cross Entropy Scores and LASER Similarity Scores for the WMT20 Parallel Corpus Filtering Shared Task

Felicia Koerner **Philipp Koehn**
Center for Language and Speech Processing
Johns Hopkins University
{fkr, phi}@jhu.edu

Abstract

This paper describes our submission to the WMT20 Parallel Corpus Filtering and Alignment for Low-Resource Conditions Shared Task. This year's corpora are noisy Khmer-English and Pashto-English, with 58.3 million and 11.6 million words respectively (English token count). Our submission focuses on filtering Pashto-English, building on previously successful methods to produce two sets of scores: LASER_LM, a combination of the LASER similarity scores provided in the shared task and perplexity scores from language models, and DCCEF_DUP, dual conditional cross entropy scores combined with a duplication penalty. We improve slightly on the LASER similarity score and find that the provided clean data can successfully be supplemented with a subsampled set of the noisy data, effectively increasing the training data for the models used for dual conditional cross entropy scoring.

1 Introduction

Machine translation systems require large amounts of high quality parallel corpora for training. Neural machine translation models in particular have been found to both require more data (Koehn and Knowles, 2017), and be more sensitive to noise in training data (Khayrallah and Koehn, 2018) than statistical machine translation models. While these data can be acquired from online sources, the resulting crawled texts are often noisy and require filtering to produce large amounts of sufficiently clean training data.

2 Related Work

We refer readers to (Koehn et al., 2019) for a more detailed overview of methods for parallel corpus filtering, here we describe the most relevant methods to this work.

2.1 Rule-based Filtering

Most filtering methods employ some rule-based filtering, usually to prepare the data for other scoring methods, based on language models, classifiers, or other translation models. (Sánchez-Cartagena et al., 2018) apply hard rules to filter out data before using a classifier to score sentence pairs. (Rossenbach et al., 2018) use many rules, including limits on sentence length, Levenshtein distance, length ratio, and token ratio. We use basic language ID and overlap rules only for the Dual Conditional Cross Entropy Scores, this is described in more detail in subsection 5.1. The LASER similarity scores provided by the shared task organizers also apply a language ID filter (assigning the pair a score of 0 if either of the sentences are not recognized as the expected language).

2.2 Dual Conditional Cross Entropy Scores

The most successful scoring method in the WMT18 Shared Task on Parallel Corpus Filtering was Dual Conditional Cross Entropy Filtering (dccef) (Junczys-Dowmunt, 2018). This method trains an NMT model in both translation directions, uses these to calculate the cross-entropy for each sentence, and finally produces a score based on their agreement. As this year's task deals with low-resource languages (contrary to WMT18, which was En-De), we explore a method to bootstrap the available clean data, thus producing more training data for the intermediate NMT models required for the method (described in more detail subsection 5.2).

2.3 LASER Similarity Scores

LASER similarity scoring was the most successful scoring method of the WMT19 Shared Task on Parallel Corpus Filtering for Low-Resource Languages (Chaudhary et al., 2019). This method embeds parallel sentences with Language Agnostic SEntence

Proceedings of the 5th Conference on Machine Translation (WMT), pages 966–971
Online, November 19–20, 2020.

Representations (LASER) (Artetxe and Schwenk, 2018), and uses these to compute cosine similarity scores. This work attempts to augment LASER similarity scores with language model scores (described in more detail in subsection 4).

3 Shared Task

For this year's shared task on Parallel Corpus Filtering and Alignment for Low-Resource conditions, participants are asked to produce scores for each of the sentence pairs in the provided noisy 58.3 million-word (English token count) Khmer-English corpus and 11.6 million-word Pashto-English corpus. These scores are used to subsample sentence pairs amounting to 5 million English words. The resulting subset is evaluated by the quality of an NMT system (`fairseq` (Ott et al., 2019)) trained on this data.

Participants were given the scripts to either train the evaluation system from scratch, or use the data to fine-tune a provided pretrained MBART model. The MBART model was trained on monolingual data, the details of which are described in (Liu et al., 2020). The performance of the NMT system is measured by BLEU score on a held-out test set of Wikipedia translations. Participants may also provide re-alignments of the source and target sentences. The organizers provide clean parallel and monolingual data for both of the language pairs, as well as LASER similarity scores, a previously successful method in low-resource conditions (Chaudhary et al., 2019), (Koehn et al., 2019).

We participated in the Pashto-English track only, after finding that the model-based methods we explored did not produce meaningful scores for Khmer-English. We did not submit sentence realignments, focusing instead on sentence filtering. Our submission builds on previously successful methods from past WMT shared tasks on parallel corpus filtering to produce two scores: LASER_LM, a combination of the LASER similarity scores and perplexity scores from language models, and DCCEF_DUP, dual conditional cross entropy scores combined with a a duplication penalty. All BLEU scores listed in this paper come from systems trained from scratch and run on the provided development data.

4 LASER_LM

A shortcoming of LASER similarity scores is that they may produce a false positive in the event that the source and target embeddings are similar to each other, but not good translations of each other. Consider, for example, a source and target pair in which the target is simply a copy of the source. This is clearly not a good translation; nothing has been translated. However, the embeddings would be exactly the same, and thus appear to be a very good match. This exact scenario is easily remedied by the use of a language identification filter, but other instances may be more difficult to root out. For example, a source and target sentence in which the target sentence is a string of literal word-for-word translations of the source sentence. To complement the LASER similarity scores and introduce some measure of fluency we train a language model for both English and Pashto.

4.1 LASER Similarity Scores

The LASER similarity scores provided are produced using the methodology outlined in the WMT19 submission (Chaudhary et al., 2019). A language identification filter is applied, and sentences pairs with an overlap between source and target of greater than 60% are discarded. The similarity scores are based on the cosine similarity between the multilingual sentence embeddings in the learned embedding space, and normalized with a margin using the k nearest neighbors approach.

4.2 Language Model Scores

Language models were trained on the provided clean monolingual data. For the English language model was trained on the Wikipedia corpus with 67,796,935 sentences. The Pashto language model was trained on a concatenation of the CommonCrawl and Wikipedia corpora, with the CommonCrawl oversampled by a factor of 64 to produce a dataset of 9,273,763 sentences. The shuffled datasets were split 90/10 (train/test) with test split into 90/10 (dev/test). The language models were trained using `fairseq` (Ott et al., 2019) with the same settings as the WikiText103 example [1].

The language model, M, was used to produce per-sentence perplexity scores for each of the sentences in the corpus. Where $s = w_1, w_2, ..., w_n$ is a sentence of length n:

$$PPL_M(s) = 2^{-\frac{1}{n}\log P(w_1,w_2,...,w_n)} \quad (1)$$

[1] https://github.com/pytorch/fairseq/blob/master/examples/language_model/README.md

scoring	BLEU (%)
LASER	9.67
LASER + 0.4 PPL_SCORE	9.82
LASER + 0.5 PPL_SCORE	9.81
LASER + 0.6 PPL_SCORE	9.62
LASER + 0.7 PPL_SCORE	9.75
LASER + 0.8 PPL_SCORE	9.88
LASER + 0.9 PPL_SCORE	9.94
LASER + 1.0 PPL_SCORE	9.57

Table 1: Results on development data (training from scratch) for different scaling factors of the PPL_SCORE.

Perplexity scores for both sides (Pashto and English, $H_{ps}(x)$ and $H_{en}(x)$ respectively) are then added together.

$$\begin{aligned}\text{PPL\_SCORE}(x) = PPL_{M_{en}}(s_{en}) \\ + PPL_{M_{ps}}(s_{ps})\end{aligned} \quad (2)$$

4.3 Combining LASER and LM Scores

The language model scores and LASER similarity scores were combined to produce LASER_LM. Both scores were normalised to fall in the range $[0, 1]$ and the PPL_SCORE subtracted from 1.0, such that lower perplexity corresponded to a higher score. Finally, the two scores were added together to produce the final score in the range $[0, 2]$. We experimented with different scaling factors f for the PPL_SCORE.

$$\begin{aligned}\text{LASER\_LM} = \text{LASER} \\ + \text{f} \cdot (1.0 - \text{PPL\_SCORE})\end{aligned} \quad (3)$$

Table 1 shows the range of factors f explored to select the scaling factor used in the final score. Since the BLEU scores produced differed only slightly, we also evaluated the models on some of the provided clean data, randomly selecting 2500 lines (roughly the size of the provided devset) from each of the clean corpora, as well as 2500 lines of a shuffled concatenation (concat) of the clean corpora. Results are shown in table 2. For the most part, they did not vary greatly, and where they did there was no consistent winner across corpora. We choose a factor of 0.5, as the model resulting from these scores generally performed well, and, importantly, performed well on the provided devset.

5 DCCEF_DUP

The dual conditional cross entropy scores produced state-of-the-art performance on the WMT18 shared task on filtering corpora for high-resource languages. However, this method requires two translation models trained in both the forward and backward direction. This presents a challenge in low-resource conditions due to the limited training data available. We find that the model quality can be improved by supplementing the provided clean data with a subsampled set consisting of 1M English tokens of the noisy data, subsampled based on the LASER similarity scores.

5.1 Preprocessing

Sentence pairs in which one or both of the sentences did not match the expected language (English or Pashto) as determined by `fastText` [2] were given a score of 0, effectively removing this pair from consideration. This is a harsh filter, removing around 45% of sentence pairs.

The resulting scores were scaled by the overlap between source and target sentence tokens, producing a sort of non-word token matching score. Note that this does not reward pairs that copy large portions of the source sentence to the target, as these are already removed by the language identification filtering.

5.2 Dual Conditional Cross Entropy Scores

Dual Conditional Cross Entropy Filtering (Junczys-Dowmunt, 2018) was found to be state of the art in the WMT18 high-resource data filtering task (Koehn et al., 2018). The method uses two translation models in the forward and backward direction, which are used to compute crosslingual similarity scores. Given the translation model M, sentence pairs (x, y) from the noisy corpus were force-decoded and a cross-entropy score produced:

$$H_M(y|x) = \frac{1}{|y|}\sum_{t=1}^{|y|} \log p_M(y_t|y_{[1,t-1],x}) \quad (4)$$

Cross-entropy scores for both directions (source-to-target and target-to-source, $H_F(y|x)$ and $H_B(x|y)$ respectively) are then averaged with a penalty on a large difference between the scores to produce the overall score:

$$\begin{aligned}\text{DCCEF}(x, y) = &\frac{H_F(y|x) + H_B(x|y)}{2} \\ &- |H_F(y|x) - H_B(x|y)|\end{aligned} \quad (5)$$

[2] https://fasttext.cc/docs/en/language-identification.html

factor	Concat	Bible	GNOME	KDE4	Tatoeba	Ubuntu	Wikimedia	TED Talks
0.4	5.84	1.79	13.08	6.98	5.21	10.73	4.65	5.76
0.5	6.28	1.07	14.09	7.72	10.31	11.02	4.83	5.17
0.6	6.76	1.62	13.89	7.41	11.39	10.36	4.34	5.43
0.7	6.43	1.18	14.02	7.98	11.02	11.42	4.28	5.21
0.8	6.20	1.00	13.71	7.87	6.66	10.92	5.11	5.71
0.9	6.25	1.80	12.99	7.32	5.80	10.32	6.02	6.15
1.0	6.67	1.63	13.77	7.44	9.53	10.75	4.54	5.69

Table 2: Results (BLEU(%)) on subsamples of clean data (training from scratch) for different scaling factors of the PPL_SCORE.

Translation models were trained using `fairseq` (Ott et al., 2019) with the same parameters used in the baseline flores model [3].

We used the provided clean training data to train translation models in both directions, and used these models to produce a dccef score as described above. Initially **only** the dccef scores were used to filter the noisy data and train a system, we did not perform the preprocessing as described in 5.1. The BLEU score produced by this system is shown in 3 under clean.

We then supplemented the clean training data with a subsample of the noisy data and trained translation models in both directions on the augmented data. The subsample of 1 million English tokens and their translations was selected based on the provided LASER similarity score. Again, for this experiment only the dccef scores were used to filter the noisy data, no preprocessing was performed. As shown in Table 3, supplementing the training data with the subsampled set resulted in an overall increase in 3.37 BLEU points.

Finally, we preprocessed the noisy data as described in 5.1 and used both sets of systems (one set trained on clean data, and one set trained on augmented data) to score the preprocessed data. As shown in Table 3, there were further, significant gains, from preprocessing, and the dccef scores from the systems trained on augmented data outperformed the dccef scores from the systems trained on just the clean data. Prepocessing also reduced the gap between the performance of dccef scores produced by systems trained on just the clean data and the performance of dccef scores produced by systems trained on augmented data.

[3] https://github.com/facebookresearch/flores#train-a-baseline-transformer-model

5.3 Duplication Penalty

The scores were scaled by a duplication penalty for duplicate (greater than one) occurrences of either one or both of the target or source sentence of a pair in the corpus as follows:

$$\text{dup\_penalty} = \begin{cases} 1.0 & \text{neither side duplicate} \\ 0.9 & \text{one side duplicate} \\ 0.8 & \text{both sides duplicate} \end{cases} \quad (6)$$

This resulted in a minor improvement in BLEU score on the development data, as seen in Table 3.

6 Results

Various other combinations of the aforementioned scores were explored, and the results are listed in Table 4. Interestingly, the results suggest that the duplication penalty did not improve the LASER_LM score, and combining the LASER_LM and DCCEF_DUP scores did not result in a better BLEU score. However, it should be noted that the differences in BLEU scores resulting from different combinations are generally minor and may not be statistically significant.

None of the filtering methods significantly outperformed the LASER-based method, but the improved dccef filtering method can at least match the LASER-based method when the training data is augmented, and the preprocessing steps and duplication penalty are applied.

7 Conclusion

This paper describes the our submission to the WMT20 Parallel Corpus Filtering Shared Task for low-resource conditions. We find that filtering based on dccef scores can compete with filtering based on LASER similarity scores when the models trained for the dccef scores are augmented with a subsample of the noisy data. This suggests that

training data for H_{en},H_{ps}	scoring method	BLEU (%)
clean	dccef	3.97
clean + top 1M noisy	dccef	7.34
clean	dccef + preprocessing	8.93
clean + top 1M noisy	dccef + preprocessing	9.68
clean + top 1M noisy	(dccef + preprocessing) · dup_penalty	**9.94**

Table 3: Results on development data (training from scratch) for dccef scores.

training data for H_{en},H_{ps} (dccef)	scoring method	BLEU (%)
N/A	laser	9.67
N/A	laser + 0.5LM	**9.81**
N/A	(laser + 0.5LM) · dup_penalty	9.74
clean	dccef	8.93
clean + top 1M noisy	dccef	9.68
clean + top 1M noisy	(dccef · dup_penalty)	**9.94**
clean + top 1M noisy	(dccef · dup_penalty) + laser	9.30
clean + top 1M noisy	(dccef · dup_penalty) + laser + 0.5LM	9.58

Table 4: Results on development data (training from scratch). Bolded scores are the two scores submitted. All dccef scores reported in this table were combined with preprocessing as described in 5.1

challenges posed by limited data for model-based filtering methods can be somewhat mitigated by bootstrapping additional data from the noisy corpus.

References

Mikel Artetxe and Holger Schwenk. 2018. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond.

Vishrav Chaudhary, Yuqing Tang, Francisco Guzmán, Holger Schwenk, and Philipp Koehn. 2019. Low-resource corpus filtering using multilingual sentence embeddings. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 261–266, Florence, Italy. Association for Computational Linguistics.

Marcin Junczys-Dowmunt. 2018. Dual conditional cross-entropy filtering of noisy parallel corpora. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 888–895, Belgium, Brussels. Association for Computational Linguistics.

Huda Khayrallah and Philipp Koehn. 2018. On the impact of various types of noise on neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 74–83, Melbourne, Australia. Association for Computational Linguistics.

Philipp Koehn, Francisco Guzmán, Vishrav Chaudhary, and Juan Pino. 2019. Findings of the WMT 2019 shared task on parallel corpus filtering for low-resource conditions. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 54–72, Florence, Italy. Association for Computational Linguistics.

Philipp Koehn, Huda Khayrallah, Kenneth Heafield, and Mikel L. Forcada. 2018. Findings of the WMT 2018 shared task on parallel corpus filtering. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 726–739, Belgium, Brussels. Association for Computational Linguistics.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39, Vancouver. Association for Computational Linguistics.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Nick Rossenbach, Jan Rosendahl, Yunsu Kim, Miguel Graça, Aman Gokrani, and Hermann Ney. 2018. The RWTH aachen university filtering system for

the WMT 2018 parallel corpus filtering task. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 946–954, Belgium, Brussels. Association for Computational Linguistics.

Víctor M. Sánchez-Cartagena, Marta Bañón, Sergio Ortiz-Rojas, and Gema Ramírez. 2018. Prompsit's submission to WMT 2018 parallel corpus filtering shared task. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 955–962, Belgium, Brussels. Association for Computational Linguistics.

Improving Parallel Data Identification using Iteratively Refined Sentence Alignments and Bilingual Mappings of Pre-trained Language Models

Chi-kiu Lo and **Eric Joanis**
Multilingual Text Processing
Digital Technologies Research Centre
National Research Council Canada (NRC-CNRC)
1200 Montreal Road, Ottawa, ON K1A 0R6, Canada
{chikiu.lo,eric.joanis}@nrc-cnrc.gc.ca

Abstract

The National Research Council of Canada's team submissions to the parallel corpus filtering task at the Fifth Conference on Machine Translation are based on two key components: (1) iteratively refined statistical sentence alignments for extracting sentence pairs from document pairs and (2) a crosslingual semantic textual similarity metric based on a pretrained multilingual language model, XLM-RoBERTa, with bilingual mappings learnt from a minimal amount of clean parallel data for scoring the parallelism of the extracted sentence pairs. The translation quality of the neural machine translation systems trained and fine-tuned on the parallel data extracted by our submissions improved significantly when compared to the organizers' LASER-based baseline, a sentence-embedding method that worked well last year. For re-aligning the sentences in the document pairs (component 1), our statistical approach has outperformed the current state-of-the-art neural approach in this low-resource context.

1 Introduction

The aim of the Fifth Conference on Machine Translation (WMT20) shared task on parallel corpus filtering (Koehn et al., 2020) is essentially the same as the two previous editions (Koehn et al., 2018b, 2019): identifying high-quality sentence pairs in a noisy corpus crawled from the web using ParaCrawl (Koehn et al., 2018a), in order to train machine translation (MT) systems on the clean data.

This year, the low-resource language pairs being tested are Khmer–English (km–en) and Pashto–English (ps–en). Specifically, participating systems must produce a score for each sentence pair in the test corpora indicating the quality of that pair. Then samples containing the top-scoring 5M words are used to train MT systems. While using the filtered parallel data to train a FAIRseq (Ott et al., 2019) neural machine translation (NMT) system remains the same as last year, the organisers are no longer building statistical machine translation (SMT) systems as part of the task evaluation. Instead, as an alternative evaluation, the filtered parallel corpus is used to fine-tune an MBART (Liu et al., 2020) pretrained NMT system. Participants were ranked based on the performance of these MT systems on a test set of Wikipedia translations (Guzmán et al., 2019), as measured by BLEU (Papineni et al., 2002). A few small sources of parallel data, covering different domains, were provided for each of the two low-resource languages. Much larger monolingual corpora were also provided for each language (en, km and ps). In addition to the task of computing quality scores for the purpose of filtering, there is also a sub-task of re-aligning the sentence pairs from the original crawled document pairs.

Cleanliness or quality of parallel corpora for MT systems is affected by a wide range of factors, e.g., the parallelism of the sentence pairs, the fluency of the sentences in the output language, etc. Previous work (Goutte et al., 2012; Simard, 2014) showed that different types of errors in the parallel training data degrade MT quality in different ways. Cross-lingual semantic textual similarity is one of the most important properties of high-quality sentence pairs. Lo et al. (2016) scored cross-lingual semantic textual similarity in two ways, either using a semantic MT quality estimation metric, or by first translating one of the sentences using MT, and then comparing the result to the other sentence, using a semantic MT evaluation metric. At the WMT18 parallel corpus filtering task, Lo et al. (2018)'s supervised submissions were developed for the same MT evaluation pipeline using a new semantic MT metric, YiSi-1 (Lo, 2019) (see also section 2.3). At the WMT19 parallel corpus filtering task, Bernier-

Proceedings of the 5th Conference on Machine Translation (WMT), pages 972–978

Colborne and Lo (2019) exploited the quality estimation metric YiSi-2 using bilingual word embeddings learnt in a supervised manner (Luong et al., 2015) from clean parallel training data or a weakly supervised manner (Artetxe et al., 2016) from bilingual dictionary. Lo and Simard (2019) further showed that using YiSi-2 with multilingual BERT (Devlin et al., 2019) on fully unsupervised parallel corpus filtering (i.e. without access of any parallel training data) achieved similar results to those in Bernier-Colborne and Lo (2019).

This year, the National Research Council of Canada (NRC) team submitted one system to the parallel corpus filtering task and one to the alignment task. The two systems share the same components in scoring the parallelism of the noisy sentence pairs, i.e., the pre-filtering rules and the quality estimation metric YiSi-2. For the parallel corpus aligning task, we use an iterative statistical alignment method to align sentences from the given document pairs before passing the aligned sentences to the scoring pipeline.

Our internal results show that MT systems trained on pre-aligned sentences filtered by our scoring pipeline outperform those trained on the organizers' LASER-based baseline (Chaudhary et al., 2019) by 0.2–1.4 BLEU. Training MT systems on re-aligned sentences using our iterative statistical alignment method achieve further gains of 0.3–1.8 BLEU.

2 System architecture

There are a wide range of factors that determine whether a sentence pair is good for training MT systems. Some of the more important properties of a good training corpus include:

- High parallelism in the sentence pairs, which affects translation adequacy.
- High fluency and grammaticality, especially for sentences in the output language, which affect translation fluency.
- High vocabulary coverage, especially in the input language, which helps make the translation system more robust.
- High variety of sentence lengths, which should also improve robustness.

In previous years, we explicitly tried to maximize all four of these properties, but this year we focused only on the first two in the scoring presented in section 2.3 below.

2.1 Iterative statistical sentence alignment

Our iterative statistical sentence alignment method as detailed in Joanis et al. (2020) uses `ssal`, a reimplementation and extension of Moore (2002) which is part of the Portage statistical machine translation toolkit (Larkin et al., 2010).

First, we train an IBM-HMM model (Och and Ney, 2003) on the clean parallel training data and the subsampled noisy corpora (see Table 1 for statistics) and use it to align paragraphs in the given document pairs, as Moore (2002) does. The subsampled noisy corpora are those obtained by applying our filtering baseline as described in sections 2.2 and 2.3 (and denoted as "nrc.baseline" in table 2). Then, we segment the paragraphs in both languages into sentences using the Portage sentence splitter. Finally, we align sentences within aligned paragraphs using the IBM model again. In this process, both the data used in training the IBM-HMM model and the noisy document pairs for alignment are punctuation tokenized using the Portage tokenizer.

In past work on sentence alignment (Joanis et al. (2020) and other unpublished experiments), we have found that first aligning paragraphs and then aligning sentences within aligned paragraphs outperforms approaches that align sentences without paying attention to paragraph boundaries.

2.2 Initial filtering

The pre-filtering steps of our submissions are mostly the same as those in Bernier-Colborne and Lo (2019). We remove:

1. duplicates after masking email, web addresses and numbers,
2. sentence pairs with a majority of number mismatches,
3. sentence pairs with either side in the wrong language according to the `pyCLD2` language detector[1],
4. sentence pairs where over half of the source sentence is non-alphabetical or target language characters, and
5. sentence pairs where over half of the target sentence is non-alphabetical characters.

An additional pre-filtering rule included in this year's submissions is the removal of pairs where over 50% of the target English sentence is directly

[1] https://github.com/aboSamoor/pycld2

Lang(s)	Training data sources	#sentence pairs	#source tokens	#target tokens
clean parallel				
km–en	JW300, Bible, GNOME/KDE/Ubuntu, Tatoeba, Global Voices	290k	6M	4M
ps–en	Bible, GNOME/KDE/Ubuntu, Wikimedia, TED Talks, Tatoeba	123k	792k	662k
filtered noisy				
km–en	ParaCrawl	288k	2M	5M
ps–en	ParaCrawl	393k	6M	5M

Table 1: Data used to train the IBM-HMM model used in the iterative statistical sentence alignment.

copying from the source Khmer or Pashto sentence.

2.3 Sentence pair scoring

The core of our sentence pair scoring component is the semantic MT quality estimation metric, YiSi-2. YiSi (Lo, 2019) is a unified semantic MT quality evaluation and estimation metric for languages with different levels of available resources. YiSi-1 measures the similarity between a machine translation and human references by aggregating weighted distributional (lexical) semantic similarities, and optionally incorporating shallow semantic structures. YiSi-2 is the bilingual, reference-less version, which uses bilingual word embeddings to evaluate cross-lingual lexical semantic similarity between the input and MT output or, in this task, between the source and target sentences.

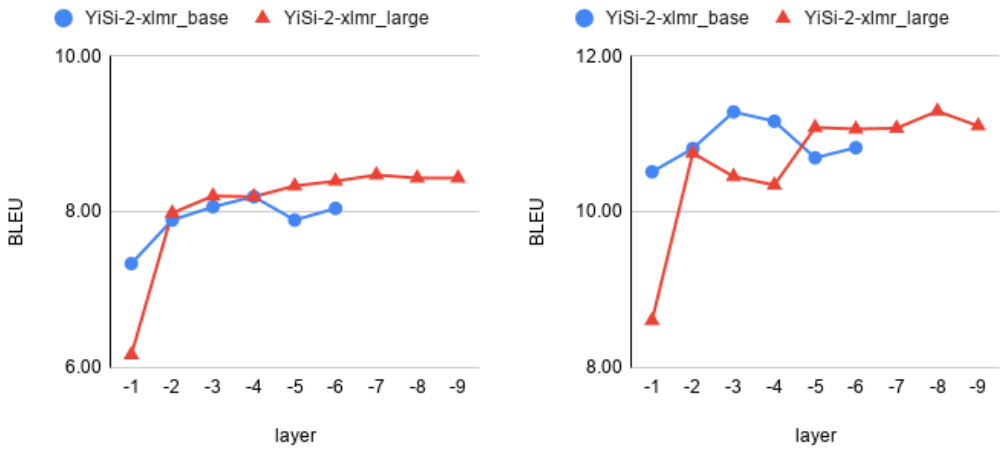
YiSi-2 relies on a crosslingal language representation to evaluate the crosslingual lexical semantic similarity. Previously, it used pre-trained multilingual BERT (Devlin et al., 2019) for this purpose. In this work, we instead experiment with XLM-RoBERTa (Conneau et al., 2020) because (1) at the time this work was done, it was the only pre-trained multilingual language encoder that covers both Khmer, Pashto and English; and (2) it shows better performance with lower-resource languages than BERT.

As suggested by Devlin et al. (2019); Peters et al. (2018); Zhang et al. (2020), we experiment with using contextual embeddings extracted from different layers of the multilingual language encoder to find out the layer that best represents the semantic space of the language.

YiSi is semantic oriented. In the past, we noticed that YiSi-based scoring functions failed to filter out sentence pairs with disfluent target text. Following Zhao et al. (2020), we experiment with improving the sentence pair scoring function by linearly combining YiSi score with the language model (LM) scores of the target text obtained from the multilingual language model used in YiSi. However, instead of using an additional pretrained language model—GPT-2 (Radford et al., 2019)—as in Zhao et al. (2020), we use the left-to-right LM scores obtained from XLM-RoBERTa while computing the crosslingual lexical semantic similarity. The advantages of using the same pretrained model for computing the crosslingual lexical semantic similarity and the language model scores are 1) it costs less in both memory and computation; 2) it is more portable to languages other than English. We combined the LM scores in the probability domain linearly with the semantic similarity scores with a weight of 0.1 assigned to the LM scores.

In the WMT19 metrics shared task (Ma et al., 2019), we saw a very significant performance degradation between YiSi-1 and YiSi-2. This suggests that current multilingual language models construct a shared multilingual space in an unsupervised manner without any direct bilingual signal, in which representations of context in the same language are likely to cluster together in part of the subspace and there is a language segregation in the shared multilingual space. Inspired by Artetxe et al. (2016) and Zhao et al. (2020), we sample 5k clean sentence pairs and use the token pairs aligned by maximum alignment of their semantic similarity to train a cross-lingual linear projection that would transform the source embeddings into the target embeddings subspace.

Lo and Larkin (2020) provide a detailed correlation analysis of YiSi-2 with all the improvements mentioned above and human judgment on



(a) km–en FAIRseq

(b) km–en MBART

(c) ps–en FAIRseq

(d) ps–en MBART

Figure 1: BLEU scores on the Khmer–English dev set for (a) FAIRseq and (b) MBART and the Pashto–English dev set for (c) FAIRseq and (d) MBART trained on 5M-word parallel subsample extracted according to the scoring functions as shown: on the x-axis, layer = $-n$ means YiSi-2 based on the embeddings of the n^{th} layer, counting from the last, of XLM-RoBERTa$_{base}$ (blue circles) or XLM-RoBERTa$_{large}$ (red triangles).

MT reference-less evaluation.

3 Experiments and results

We used the software provided by the task organizers to extract the 5M-word samples from the original test corpora according to the scores generated by each alignment and/or filtering system. We then trained a FAIRseq MT system or fine-tuned an MBART pretrained NMT using the extracted subsamples. The MT systems were then evaluated on the official dev set ("dev-test").

We exhaustively experimented with the last few layers of both XLM-RoBERTa$_{base}$ and XLM-RoBERTa$_{large}$ in order to find out the model and layer best representing crosslingual semantic similarity. Figure 1 shows the plots of the change in BLEU scores of each MT system using the embeddings extracted from the n^{th} layer, counting from the last, of the multilingual LM for evaluating crosslingual lexical semantic similarity. In general, we see a trend of rising performance as we roll back from the last layer. The performance peaks at some point and starts to fall when we roll back too far from the end. For XLM-RoBERTa$_{base}$, the peak performance of the MT systems is achieved by the 3$^{\text{rd}}$ or 4$^{\text{th}}$ last layer (out of 12 layers). For XLM-RoBERTa$_{large}$, the peak performance of the MT systems is achieved by the 8$^{\text{th}}$ last layer (out of 24 layers). The peak performance of MT systems trained on sentences filtered by XLM-RoBERTa$_{large}$ based YiSi-2 is better than that by XLM-RoBERTa$_{base}$ based YiSi-2.

system	alias	km–en		ps–en	
		FAIRseq	MBART	FAIRseq	MBART
filtering only					
LASER	baseline	7.10	10.13	9.77	11.03
+ filter rules		7.55	10.44	9.87	11.91
YiSi-2-xlmr_large (layer -8) + filter rules	nrc.baseline	8.43	11.29	**9.96**	12.54
+ LM score		8.53	11.31	9.61	**12.82**
+ LM score + CLP_{5k}	nrc.filtering	**8.54**	**11.58**	9.93	12.80
re-aligning and filtering					
iterative alignment + nrc.filtering	nrc.alignment	***8.82***	11.17	***11.73***	***13.21***

Table 2: BLEU scores of selected systems. The two final submitted systems are labelled nrc.filtering and nrc.alignment.

Table 2 shows the results of the experiments described in section 2.3. First, we show an improved version of the organizers' baseline by simply adding our initial filtering rules. This shows that our initial filtering rules are able to catch bad parallel sentences which are hard to filter by an embedding-based filtering system.

Next, we see that using YiSi-2 with XLM-RoBERTa$_{large}$'s 8th last layer as parallelism scoring function outperforms the LASER baseline by 0.1–0.9 BLEU in different translation directions and MT architectures. This is our "nrc.baseline" system, and the baseline used for filtering the noisy corpus in training the IBM-HMM alignment model for the "nrc.alignment" system. Adding the LM score to the scoring function shows small improvements. Learning the cross-lingual linear projection matrix to transform the source embeddings in the target language subspace shows more improvements overall. This is our "nrc.filtering" submission to the parallel corpus filtering task.

At last, we show that using our iterative statistical alignment method to redo the alignment of sentences from the given document pairs improves the translation quality of the resulting MT systems significantly. This is our "nrc.alignment" submission to the parallel corpus filtering task.

4 Conclusion and Future Work

In this paper, we presented the NRC's two submissions to the WMT20 Parallel Corpus Filtering and Alignment for Low-Resource Conditions task. Our experiments show that YiSi-2 is a scoring function of parallelism that is very competitive, and that a statistical sentence alignment method is still able to provide better alignment results than neural ones in low resource situations. Further analysis is required to understand the characteristics of the sentence pairs aligned by the baseline vecalign and our iterative statistical sentence alignment and how the latter achieves better translation quality for the trained MT systems.

It is worth highlighting that in this task, as well as in our Inuktitut–English corpus alignment work (Joanis et al., 2020), a well-tuned statistical sentence-alignment system outperformed a state-of-the-art neural one. We hypothesise that this is a low-resource effect, but further work is still needed to explore the best low-resource corpus alignment methods. In particular, we intend to integrate YiSi-2 into our sentence aligner to test whether it's our iterative alignment methodology that makes the difference or the fact that the underlying scoring function is statistical (we use IBM-HMM models for sentence pair scoring in our aligner). It's possible that the statistical approach might continue to win here, because in the low-resource context there might not be enough training data to tune the orders of magnitude more parameters of the neural models; a counter-argument is that YiSi-2 did better on the scoring task than statistical scoring functions. Our future work will explore the trade-offs between these two approaches, and consider hybrid methods.

Acknowledgements

We thank Samuel Larkin and Marc Tessier for their help in setting up the FAIRseq and MBART baselines using the LASER scores; and Patrick Littell for discussion and feedback on the Pashto test set. We also thank the reviewers for their comments and suggestions, and Roland Kuhn for his comments and feedback on the paper.

References

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2016. Learning principled bilingual mappings of word embeddings while preserving monolingual invariance. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2289–2294, Austin, Texas. Association for Computational Linguistics.

Gabriel Bernier-Colborne and Chi-kiu Lo. 2019. NRC parallel corpus filtering system for WMT 2019. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 252–260, Florence, Italy. Association for Computational Linguistics.

Vishrav Chaudhary, Yuqing Tang, Francisco Guzmán, Holger Schwenk, and Philipp Koehn. 2019. Low-resource corpus filtering using multilingual sentence embeddings. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 261–266, Florence, Italy. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Cyril Goutte, Marine Carpuat, and George Foster. 2012. The impact of sentence alignment errors on phrase-based machine translation performance. In *Proceedings of the Tenth Conference of the Association for Machine Translation in the Americas*.

Francisco Guzmán, Peng-Jen Chen, Myle Ott, Juan Pino, Guillaume Lample, Philipp Koehn, Vishrav Chaudhary, and Marc'Aurelio Ranzato. 2019. The FLORES evaluation datasets for low-resource machine translation: Nepali–English and Sinhala–English. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6098–6111, Hong Kong, China. Association for Computational Linguistics.

Eric Joanis, Rebecca Knowles, Roland Kuhn, Samuel Larkin, Patrick Littell, Chi-kiu Lo, Darlene Stewart, and Jeffrey Micher. 2020. The Nunavut Hansard Inuktitut–English parallel corpus 3.0 with preliminary machine translation results. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 2562–2572, Marseille, France. European Language Resources Association.

Philipp Koehn, Vishrav Chaudhary, Ahmed El-Kishky, Naman Goyal, Peng-Jen Chen, and Francisco Guzmán. 2020. Findings of the WMT 2020 shared task on parallel corpus filtering and alignment. In *Proceedings of the Fifth Conference on Machine Translation: Shared Task Papers*.

Philipp Koehn, Francisco Guzmán, Vishrav Chaudhary, and Juan Pino. 2019. Findings of the WMT 2019 shared task on parallel corpus filtering for low-resource conditions. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 54–72, Florence, Italy. Association for Computational Linguistics.

Philipp Koehn, Kenneth Heafield, Mikel L. Forcada, Miquel Esplà-Gomis, Sergio Ortiz-Rojas, Gema Ramírez Sánchez, Víctor M. Sánchez Cartagena, Barry Haddow, Marta Bañón, Marek Střelec, Anna Samiotou, and Amir Kamran. 2018a. ParaCrawl corpus version 1.0. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Philipp Koehn, Huda Khayrallah, Kenneth Heafield, and Mikel Forcada. 2018b. Findings of the WMT 2018 shared task on parallel corpus filtering. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, Brussels, Belgium. Association for Computational Linguistics.

Samuel Larkin, Boxing Chen, George Foster, Ulrich Germann, Eric Joanis, Howard Johnson, and Roland Kuhn. 2010. Lessons from NRC's Portage system at WMT 2010. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 127–132, Uppsala, Sweden. Association for Computational Linguistics.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation.

Chi-kiu Lo. 2019. YiSi - a unified semantic MT quality evaluation and estimation metric for languages with different levels of available resources. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 507–513, Florence, Italy. Association for Computational Linguistics.

Chi-kiu Lo, Cyril Goutte, and Michel Simard. 2016. CNRC at SemEval-2016 task 1: Experiments in crosslingual semantic textual similarity. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 668–673, San

Diego, California. Association for Computational Linguistics.

Chi-kiu Lo and Samuel Larkin. 2020. MT reference-less evaluation using YiSi-2 with bilingual mappings of massive multilingual language model. In *Proceedings of the Fifth Conference on Machine Translation: Shared Task Papers*.

Chi-kiu Lo and Michel Simard. 2019. Fully unsupervised crosslingual semantic textual similarity metric based on BERT for identifying parallel data. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 206–215, Hong Kong, China. Association for Computational Linguistics.

Chi-kiu Lo, Michel Simard, Darlene Stewart, Samuel Larkin, Cyril Goutte, and Patrick Littell. 2018. Accurate semantic textual similarity for cleaning noisy parallel corpora using semantic machine translation evaluation metric: The NRC supervised submissions to the parallel corpus filtering task. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 908–916, Belgium, Brussels. Association for Computational Linguistics.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Bilingual word representations with monolingual quality in mind. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 151–159, Denver, Colorado. Association for Computational Linguistics.

Qingsong Ma, Johnny Wei, Ondřej Bojar, and Yvette Graham. 2019. Results of the WMT19 metrics shared task: Segment-level and strong MT systems pose big challenges. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 62–90, Florence, Italy. Association for Computational Linguistics.

Robert C. Moore. 2002. Fast and accurate sentence alignment of bilingual corpora. In *Proceedings of the Conference of the Association for Machine Translation in the Americas*, pages 135–144.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, Philadelphia, Pennsylvania, USA.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9.

Michel Simard. 2014. Clean data for training statistical MT: the case of MT contamination. In *Proceedings of the Eleventh Conference of the Association for Machine Translation in the Americas*, pages 69–82, Vancouver, BC, Canada.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with BERT. In *International Conference on Learning Representations*.

Wei Zhao, Goran Glavaš, Maxime Peyrard, Yang Gao, Robert West, and Steffen Eger. 2020. On the limitations of cross-lingual encoders as exposed by reference-free machine translation evaluation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1656–1671, Online. Association for Computational Linguistics.

Alibaba Submission to the WMT20 Parallel Corpus Filtering Task

Jun Lu, Xin Ge, Yangbin Shi, Yuqi Zhang
Machine Intelligence Technology Lab, Alibaba Group
Hangzhou, China
{joelu.luj, shiyi.gx, taiwu.syb, chenwei.zyq}@alibaba-inc.com

Abstract

This paper describes the Alibaba Machine Translation Group submissions to the WMT 2020 Shared Task on Parallel Corpus Filtering and Alignment. In the filtering task, three main methods are applied to evaluate the quality of the parallel corpus, i.e. a) Dual Bilingual GPT-2 model, b) Dual Conditional Cross-Entropy Model and c) IBM word alignment model. The scores of these models are combined by using a positive-unlabeled (PU) learning model and a brute-force search to obtain additional gains. Besides, a few simple but efficient rules are adopted to evaluate the quality and the diversity of the corpus. In the alignment-filtering task, the extraction pipeline of bilingual sentence pairs includes the following steps: bilingual lexicon mining, language identification, sentence segmentation and sentence alignment. The final result shows that, in both filtering and alignment tasks, our system significantly outperforms the LASER-based system.

1 Introduction

The parallel corpus is an essential resource for building a high quality machine translation(MT) system. It has been shown that, the higher the corpus quality, the better the performance of a MT system(Koehn and Knowles, 2017; Khayrallah and Koehn, 2018). Many successful machine translation systems are built on the corpus crawled from the web. In practice, this kind of parallel corpus may be very noisy. The task of Parallel Corpus Filtering is aimed at tackling the problem of cleaning noisy parallel corpora.

We form the bilingual sentences quality in the following aspects. Firstly, a *high-quality* parallel sentence pair(also called bitext) should have the property that its target sentence precisely translates the source sentence, and vice versa. In this task, we attempt to quantify the translation accuracy (also called bilingual score) of bilingual sentence pairs. Secondly, the monolingual quality of the target and source sentences of a parallel corpus should also be considered. In our system, we evaluate the monolingual quality (also called monolingual score) of a target sentence due to its importance for the MT procedure. Finally, the bilingual and monolingual scores are combined to evaluate bilingual sentence pairs and filter out the ones with low quality.

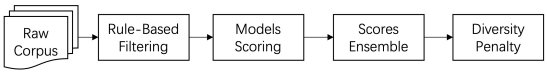


Figure 1: Framework of parallel corpus filtering

The paper is structured as follows. Section 2 describes our methods which are used in the parallel corpus filtering. In Section 3, we briefly outline the pipeline of parallel sentence extraction. Section 4 specifies the experiments and results as well as the dataset for building model-based methods. Conclusions are drawn in Section 5.

2 Parallel Corpus Filtering Methods

Figure 1 shows the framework of parallel corpus filtering. The raw parallel corpus is firstly filtered by heuristic rules so that the very noisy sentence pairs will be removed. Then, the bilingual & monolingual models are built to score all the remaining sentence pairs. By using an ensemble model, the partial scores of each sentence pair are combined to a single quality score.

2.1 Rule-based Filtering

A series of heuristic rules(Lu et al., 2018) are applied to filter low quality sentence pairs. They are simple, (almost) language independent but efficient, which are described below.

Proceedings of the 5th Conference on Machine Translation (WMT), pages 979–984
Online, November 19–20, 2020.

Monolingual Rules

- The length of the sentence which is too short (≤ 2 tokens) or too long (> 200 tokens) will be dropped. In our system, sentences(English, Khmer and Pashto) are tokenized by SentencePiece[1].

- The ratio of the valid tokens count to the length of the sentence. Here, valid tokens are the ones which contain the letters in the corresponding language. For example, a valid token in English should contain English letters. In our system, the sentence is filtered out if its valid-tokens ratio is less than 0.2.

- Language filtering. For the Pashto-English parallel corpus, the languages of source and target sentences should be Pashto and English. We detect the language of a sentence by using a language detection tool we developed[2]. A sentence pair is dropped when its source language and target language are not Pashto and English, respectively.

Bilingual Rules

- The length ratio of a source sentence to a target sentence. The sentence length is calculated by the number of sentencepiece tokens. In our system, the ratio is set between 0.2 and 5.0 for both language pairs.

- The edit distance between the source token sequence and the target token sequence. A small edit distance indicates that the source and target sentences are very similar, which harms the performance of the NMT system a lot (Khayrallah and Koehn, 2018).

- The consistency of special tokens (Taghipour et al., 2010). For example, the high-quality sentence pairs should contain the same email address in both source and target sentences (if exists). In this task, special tokens are email addresses, URLs, and big Arabic numbers.

2.2 Dual Bilingual GPT-2 Model

Inspired by the *Cross-lingual Language Model Pretraining* work of (Lample and Conneau, 2019), we propose a Translation Language Model(called Bilingual GPT-2 model) based on the GPT-2 model(Radford et al., 2019). As illustrated in Figure 2, the Bilingual GPT-2 model is trained with both monolingual and parallel sentences. For parallel sentence pairs, we concatenate the source and target sides to obtain a long sentence and then feed it to the model. For monolingual sentences, we convert them to *fake sentences pairs* by assigning the corresponding side sentence with a unique token. For example, when an English sentence "Hello word." is used in the English-Khmer bilingual GPT-2 model training, a fake sentence pair, ("Hello word", "<KM>"), will be used. Here, the English sentence is the source and "<KM>" is the target. While training, a large number of fake bilingual corpora are firstly used to *pre-train* the model. Then, the real clean parallel sentence pairs are used to *fine-tune* the model. In this task, we trained two Bilingual GPT-2 models for each language pair, i.e., source-to-target and target-to-source models. The two translation quality scores from the Dual Bilingual GPT-2 model are given precisely by:

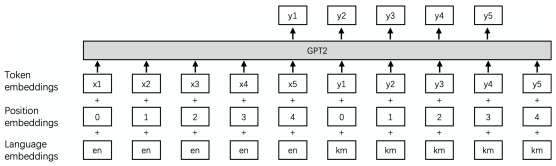


Figure 2: Bilingual GPT-2 model structure

$$score_1(x,y) = \frac{1}{2}\Big(\sum_{t\in|y|} \log p_{s2t}(y_t) + \sum_{t\in|x|} \log p_{t2s}(x_t)\Big) \quad (1)$$

$$score_2(x,y) = \frac{1}{2}\Big(\sum_{t\in|y|} \log p_{s2t}(y_t) - \log p_{t2s}(y_t) + \sum_{t\in|x|} \log p_{t2s}(x_t) - \log p_{s2t}(x_t)\Big) \quad (2)$$

In Equation (1) and (2), x and y are the source and target sentences. $logp_{s2t}(y_t)$ represents the cross-entropy loss of the target side token y_t, which is obtained by the source-to-target model. $\log p_{t2s}(x_t)$ represents the cross-entropy loss of the source side token x_t, which is obtained from the target-to-source model.

We don't use the BERT model here, as it is hard for computing the cross-entropy loss efficiently.

[1] https://github.com/google/sentencepiece

[2] This tool is similar to Google's CLD2: https://github.com/CLD2Owners/cld2

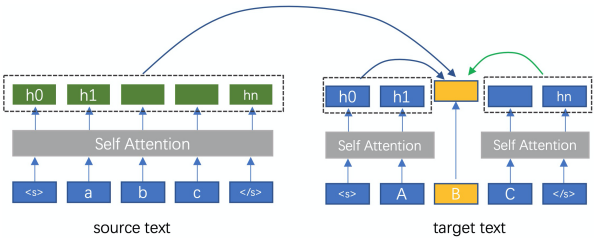


Figure 3: Optimized cross entropy model (source-to-target)

2.3 Dual Conditional Cross-Entropy Model

The dual conditional cross-entropy model(Junczys-Dowmunt, 2018) has been proven effective in parallel corpus filtering, which uses a combination of forward and backward models to compute a force-decoding score. In our system, the model is optimized to better evaluate the quality of the parallel sentences in low-resource languages.

Specifically, Figure 3 shows the structure of our model. Each token in a target side sentence is predicted by its left and right context and the source text. Hence, the cross-entropy score of a sentence pair is given below:

$$H_M(y|x) = \frac{1}{|y|} \sum_{t \in |y|} \log p_M(y_t|x, y_{<t}, y_{>t})$$

The final bilingual quality score combines the source to target and target to source cross-entropy scores as below:

$$score(x,y) = \frac{1}{2}(H_{Fwd}(y|x) + H_{Bck}(x|y)) + |H_{Fwd}(y|x) + H_{Bck}(x|y)|$$

As shown in Figure 3, the source sentence and target context are encoded by two 12-layer transformer models with hidden size 768. In fact, the target side model can be regarded as a bidirectional GPT-2 model. In our system, the source and target side transformer models are pre-trained by using large amount of monolingual data. Then, the models are fine-tuned by clean bilingual sentences pairs.

2.4 IBM Word Alignment Model

The word alignment model can be used for evaluating the translation quality of bilingual sentence pairs (Khadivi and Ney, 2005; Taghipour et al., 2010; Ambati, 2011). Inspired by the work of (Khadivi and Ney, 2005), we simplify the original algorithm, and the translation score of sentence pairs is given below:

$$score(s,t) = \frac{1}{|s|} \sum_{s_i,t_j \in a_{s2t}} \log p(t_j|s_i) + \frac{1}{|t|} \sum_{s_i,t_j \in a_{t2s}} \log p(s_i|t_j) \quad (3)$$

In Equation (3), s and t represent the source and target sentences respectively, $p(w_1|w_2)$ indicates the word translation probability, and a_{s2t} indicates the source words to target words alignment.

In this task, by using the *fast_align* toolkit (Dyer et al., 2013), the word alignment model is trained on a clean parallel corpus as described in Section 4.1 to get the forward and reverse word translation probability tables. This model is also called alignment scoring model.

2.5 GPT-2 Language Model

In this task, GPT-2 language model is applied to compute the monolingual scores of source and target sentences. We train GPT-2 models for each language by using the *HuggingFace Transformers* toolkit (Wolf et al., 2019) with the monolingual data provided by the task organizers. The training data is cleaned by the rules described in the Section 2.1. The configuration of the GPT-2 model is also the same with the *GPT2-large model* described in the work of (Radford et al., 2019).

2.6 Ensemble

Each sentence pair in the noisy parallel corpus is scored by each of the models described above. As a result, each sentence pair would obtain a few partial scores. We need a single score based on the partial scores to rank the sentence pairs.

At first, we turn the scores from each model to the values between 0 and 1. Specifically, the scores are normalized with the method described in (Junczys-Dowmunt, 2018), which is based on the entropy information.

Then, a single score $f(x,y)$ is produced as the product of partial scores $f_i(x,y)$. Since the different importance of the partial scores, the lower boundary value of the scores is represented as θ, where $0 \leq \theta \leq 1$, which results in a new normalization range $[\theta, 1]$. The more important the model is, the closer to 0 the θ is. It means that the scores from this model could distribute from 0 to 1, which

would affect a lot on the final score. On the contrary, when we set θ close to 1, the model has minor impact on the final score whatever its distribution is. Hence, the single score is given by:

$$f(x,y) = \prod_i f_i(x,y), f_i(x,y) \in [\theta_i, 1] \quad (4)$$

We applied the brute-force search to find the best θs for the models. Compared to the pure production of the partial scores, our method has improved 0.5% - 1.0% BLEU score(Papineni et al., 2002).

In addition, the ensemble could also be treated as a Positive-Unlabeled classification task (Chaudhary et al., 2019). We use the officially released high quality data and the sentence pairs which are ranked top by our models mentioned above as the positive samples. Meanwhile, the sentence pairs from the noisy parallel corpus are treated as the unlabeled samples. As a result, the PU classification based on the random forest models has contributed 0.1% - 0.2% improvement on the development data.

In our final submissions, the brute-force search method and PU-classification are used in Khmer-English and Pashto-English filtering tasks respectively.

3 Pipeline of Parallel Corpus Extraction

Bilingual Lexicon Extraction. In the first step, by using the word alignment model, parallel token pairs are extracted from the clean parallel corpus. Specifically, after tokenization, the parallel corpus is fed to the fast align toolkit to obtain the mutual translation probabilities dictionary. We then extract the token pairs with forward and backward translation probabilities higher than 0. The bilingual lexicon (i.e., the collection of parallel token pairs) will iteratively be updated after more bitexts are mined, since the lexicon is the cornerstone of bitexts mined from aligned documents which is described below.

Language Identification. The second step is to identify the language of each document by using a language detection tool we developed. In this way, a document pair will be discarded if its detection results do not match the expected languages.

Sentence Segmentation. This step is to split sentences in documents with rules or models. A few rules based on end-of-sentence punctuations are used to split sentences of language Pashto and Khmer. For English sentence segmentation, a segmentation model is built via nltk toolkit[3].

Language	Sentences	English Words
Khmer-English	270K	4.2M
Pashto-English	106k	1.9M

Table 1: Clean bitexts used in bilingual models training

Sentence Alignment. In this step, a dynamic programming framework based on bilingual lexicon (Ma, 2006) is built to mine parallel sentence pairs.

Corpus Filtering. Finally, the extracted bitexts are cleaned by using the methods described in section 2. And as mentioned above, we mix the new mined bitexts with the provided bitexts from WMT2020 to iteratively run the fast_align model to update bilingual lexicon.

4 Experiments and Results

In this section, we specify the experimental settings and results in the corpus filtering and alignment task.

4.1 Corpora and Settings

The selection data pool[4] (which we called noisy dataset) is provided by *WMT20 Corpus Filtering and Alignment Task*. It contains 1.02 million sentences pairs of Pashto-English corpus and 4.17 million sentences pairs of Khmer-English corpus. These parallel corpora are very noisy. The task's participants are asked to sub-select sentence pairs that amount to 5 million English words for each of the noisy parallel sets. The quality of the resulting subsets is determined by the BLEU scores of a neural machine translation system[5] trained on selected data. In our NMT experiments, we use the NMT configuration that is provided by the task organizers[6] as well as the development and test sets.

In addition, organisers provide the permissible third-party sources of parallel corpora, which we called "official parallel data". Additional monolingual corpora are also provided for English, Khmer and Pashto languages. For sentence pair alignment task, the organisers also provide the document pairs

[3]Natural Language Toolkit: https://github.com/nltk/nltk

[4]http://www.statmt.org/wmt20/parallel-corpus-filtering.html

[5]https://github.com/pytorch/fairseq.git

[6]http://data.statmt.org/wmt20/filtering-task/dev-tools.tgz

Method	km-en			ps-en		
	pairs counts ($\times 10^4$)	normal train	finetune	pairs counts ($\times 10^4$)	normal train	finetune
LASER(Baseline)	24.1	7.35	10.4	22.5	9.66	10.76
LASER +Rules	24.7	7.56	10.89	22.9	9.88	11.13
IBM word align +Rules	25.7	8.25	11.04	35.6	10.37	12.49
Dual X-Ent +Rules	34.6	8.12	10.71	43.4	9.9	12.14
Dual bi-GPT-2_1 +Rules	33.3	8.3	10.86	30.2	9.95	11.62
Dual bi-GPT-2_2 +Rules	38.1	8.5	10.95	34.9	10.04	12.17
Ensemble + Rules	25.8	8.61	11.34	37.5	10.84	12.75
Alignment + Ensemble + Rules	-	-	-	21.2	11.36	13.29

Table 2: Main results for corpus filtering and alignment task

in which the participants can extract bilingual sentence pairs.

In Section 2, we introduced 3 sub-models for translation quality scoring, i.e. Dual Bilingual GPT-2 Model, Dual Conditional Cross-Entropy and IBM Word Alignment Model. These models can be trained with the monolingual and clean parallel corpus. In particular, the clean parallel data is more important in training. Unfortunately, both Khmer-English and Pashto-English are low-resource language pairs and lack parallel corpus. Therefore, in order to expand the clean parallel dataset, the high quality sentence pairs are selected/extracted from the noisy dataset or the parallel document pairs by using an iterative process in our filtering system. Specifically, the corpus filtering models are initially trained by using the official parallel data. Then, these models are used to estimate the quality of the sentence pairs in the noisy dataset and parallel documents. Finally, by applying some rules and strict threshold value, the high quality sentence pairs are selected and combined with official parallel data to train the new version of filtering models. The process described above was repeated 3 times and achieved larger clean parallel corpora as detailed in Table 1.

For text preprocessing, we built two joint SentencePiece models for Khmer-English and Pashto-English respectively with the 60k vocabulary size. Then, monolingual and bilingual texts are tokenized by the corresponding SentencePiece models.

4.2 Experimental Results

Our main results are shown in Table 2. All NMT experiments were done in the same environment with 2 GPUs for normal training (i.e., NMT training from scratch) and 1 GPU for MBART-based fine-tuning[7]. The LASER scores provided by the organisers were used as baseline scores, which achieved reasonable results in both normal training and MBART-based fine-tuning. Our rules proposed above were firstly used to filter very noisy sentence pairs and achieved a slightly better performance. Then, the 3 main bitexts scoring models were combined with rules respectively to test their effectiveness in experiments. We found that, the IBM word alignment model was reliable in most cases and Dual Bilingual GPT-2 model slightly outperformed the Dual Conditional Cross-Entropy model. Finally, the ensemble model obtained the highest BLEU scores in the filtering task.

In the task of sentence pairs alignment, we only submitted the results of Pashto-English. While extracting sentence pairs, 13,976 bilingual word pairs were firstly obtained from the clean parallel corpus. As a result, we mined 723,414 sentence pairs from 45,307 document pairs and achieved an improvement of 0.5 BLEU score.

5 Conclusions

In this paper, we present our corpus filtering system for the *WMT 2020 Corpus Filtering Task*. In our system, Dual Bilingual GPT-2 model, Dual Conditional Cross-Entropy model and IBM word alignment model are combined to filter the noisy parallel corpus. Besides, a parallel sentence pairs extraction system is built to re-align the bilingual sentences. The experiments show that, compared

[7]The MBART pre-trained models were provided by the organizers and described here, http://www.statmt.org/wmt20/parallel-corpus-filtering.html.

to the baseline system, our filtering and extraction system achieve much better results.

Acknowledgments

This work is supported by National Key RD Program of China (2018YFB1403202).

References

Vamshi Ambati. 2011. *Active learning and crowdsourcing for machine translation in low resource scenarios*. Ph.D. thesis, University of Southern California.

Vishrav Chaudhary, Yuqing Tang, Francisco Guzmán, Holger Schwenk, and Philipp Koehn. 2019. Low-resource corpus filtering using multilingual sentence embeddings. In *Proceedings of the Fourth Conference on Machine Translation, WMT 2019, Florence, Italy, August 1-2, 2019 - Volume 3: Shared Task Papers, Day 2*, pages 261–266. Association for Computational Linguistics.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648, Atlanta, Georgia. Association for Computational Linguistics.

Marcin Junczys-Dowmunt. 2018. Dual conditional cross-entropy filtering of noisy parallel corpora. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 901–908, Belgium, Brussels. Association for Computational Linguistics.

Shahram Khadivi and Hermann Ney. 2005. Automatic filtering of bilingual corpora for statistical machine translation. In *International Conference on Application of Natural Language to Information Systems*, pages 263–274. Springer.

Huda Khayrallah and Philipp Koehn. 2018. On the impact of various types of noise on neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 1–10.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. *arXiv preprint arXiv:1706.03872*.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *Advances in Neural Information Processing Systems (NeurIPS)*.

Jun Lu, Xiaoyu Lv, Yangbin Shi, and Boxing Chen. 2018. Alibaba submission to the wmt18 parallel corpus filtering task. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 930–935, Belgium, Brussels. Association for Computational Linguistics.

Xiaoyi Ma. 2006. Champollion: A robust parallel text sentence aligner. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy. European Language Resources Association (ELRA).

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9.

Kaveh Taghipour, Nasim Afhami, Shahram Khadivi, and Saeed Shiry. 2010. A discriminative approach to filter out noisy sentence pairs from bilingual corpora. In *Telecommunications (IST), 2010 5th International Symposium on*, pages 537–541. IEEE.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Volctrans Parallel Corpus Filtering System for WMT 2020

Runxin Xu[1,3], **Zhuo Zhi**[2,3], **Jun Cao**[3], **Mingxuan Wang**[3], **Lei Li**[3]
[1] Institute of Computational Linguistic, Peking University
[2] Department of Computer Science and Engineering, University of California San Diego
[3] ByteDance AI Lab, Shanghai, China
runxinxu@gmail.com
zzhi@ucsd.edu
{caojun.sh, wangmingxuan.89, lileilab}@bytedance.com

Abstract

In this paper, we describe our submissions to the WMT20 shared task on parallel corpus filtering and alignment for low-resource conditions. The task requires the participants to align potential parallel sentence pairs out of the given document pairs, and score them so that low-quality pairs can be filtered. Our system, Volctrans, is made of two modules, i.e., a mining module and a scoring module. Based on the word alignment model, the mining module adopts an iterative mining strategy to extract latent parallel sentences. In the scoring module, an XLM-based scorer provides scores, followed by reranking mechanisms and ensemble. Our submissions outperform the baseline by 3.x/2.x and 2.x/2.x for km-en and ps-en on From Scratch/Fine-Tune conditions.

1 Introduction

With the rapid development of machine translation, especially **N**eural **M**achine **Translation** (NMT) (Vaswani et al., 2017; Ott et al., 2018; Zhu et al., 2020), parallel corpus in high quality and large quantity is in urgent demand. These parallel corpora can be used to train and build robust machine translation models. However, for some language pairs on low-resource conditions, few parallel resources are available. Since it is much easier to obtain quantities of monolingual data, it may help if we can extract parallel sentences from monolingual data through alignment and filtering.

The WMT19 shared task on parallel corpus filtering for low-resource conditions (Koehn et al., 2019) provides noisy parallel corpora in Sinhala-English and Nepali-Englis crawled from the web. Participants are asked to score sentence pairs so that low-quality sentences are filtered. In this year, the WMT20 shared task on parallel Corpus filtering and alignment for low-resource conditions is very similar, except that the language pairs become Khmer-English and Pashto-English, and the provided raw data are documents in pair, which require sentence-level alignment. Besides, no data in similar languages are provided for this year.

The participants are required to align sentences within documents in different languages and provide a score for each sentence pair. To evaluate the quality of the extracted sentence pairs, they are subsampled to 5 million English words and used to train a neural machine translation model. Finally, the BLEU score of the machine translation system is used to reflect the quality of the sentence pairs.

In this paper, we propose the Volctrans filtering system, which consists of a **mining module** and a **scoring module**. First, the mining module extracts and aligns potential parallel sentence pairs within documents in different languages. In particular, we introduce an iterative mining strategy to boost mining performance. We keep adding newly aligned high-quality parallel sentences to train the word alignment model, which is essential for the mining module. Second, the scoring module is based on XLM (Conneau and Lample, 2019), and responsible for providing scores for each sentence pair. Several reranking mechanisms are also used in this module. We conduct experiments to tune the hyper-parameters for the best filtering performance, and four systems are ensembled to achieve the final results.

2 System Architecture

2.1 Data Introduction

In detail, as is shown in Table 1, the WMT20 shared task provides:

- Document pairs, including $391,250$ Khmer-English and $45,312$ Pashto-English document pairs;
- Sentence-aligned corpora extracted from the

Proceedings of the 5th Conference on Machine Translation (WMT), pages 983–990
Online, November 19–20, 2020.

above document pairs, using Hunalign (DBL, 2008) and LASER (Artetxe and Schwenk, 2019), including $4,169,574$ Khmer-English and $1,022,883$ Pashto-English sentence pairs;

- Parallel data which can be used to build filtering and alignment system, including $290,051$ Khmer-English and $123,198$ Pashto-English parallel sentences;
- Monolingual data, including approximately 1.9 billion English, 14.0 million Khmer, and 6.6 million Pashto sentences.

Table 1: Statistics of Provided Data Scale

	en-km	en-ps
Document Pairs	391K	45K
Extracted Sentence Pairs	4.2M	1.0M
Parallel Sentences	290K	123K

2.2 Mining Module

Besides the given aligned sentences, we believe that there are still more potential parallel sentences that can be mined. Thus we choose to extract our own set of sentence pairs from the provided document pairs, and design a mining module aiming to gather as many parallel sentence candidates as possible. We then elaborate on our mining procedure and mining module, shown in Figure 1, in detail.

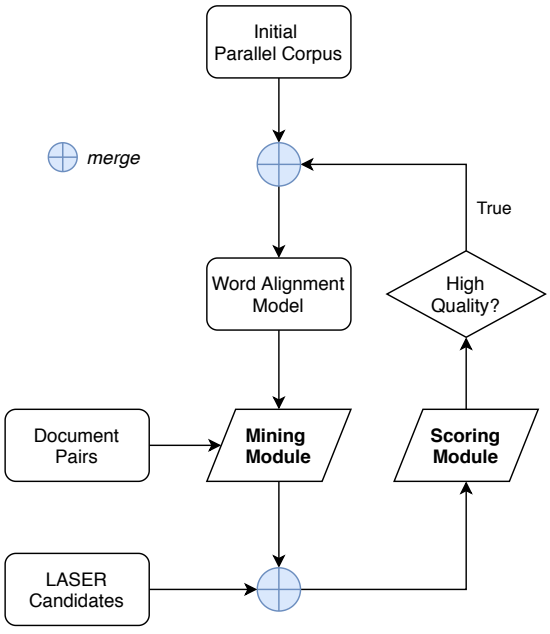


Figure 1: Mining Procedure

2.2.1 Word Alignment

We trained the word alignment model on the provided clean parallel corpus by using the *fast-align* toolkit (Dyer et al., 2013), and get the forward and reverse word translation probability tables. It's worth mentioning that both of Pashto and Khmer corpus are tokenized before word alignment model training for accuracy consideration. We separate Pashto words by moses tokenizer[1]. For Khmer, we use the character (\u200B in Unicode) as separator when it's available and otherwise use a dictionary-based tokenizer by maximizing the word sequence probability.

2.2.2 Mining Parallel Sentences

This step is operated by our mining module. With the bilingual word translation probability tables, the mining module evaluates the translation quality of bilingual sentence pairs by YiSi-2 (Lo, 2019), which involves both lexical weight and lexical similarity. The Document pairs are first segmented on each language side using Polyglot [2]. This initial segmentation is represented as:

$$e = \overline{e}_1 \overline{e}_2 \cdots \overline{e}_a = \overline{e}_1^a \quad (1)$$

$$f = \overline{f}_1 \overline{f}_2 \cdots \overline{f}_b = \overline{f}_1^b \quad (2)$$

where $\overline{e}_k$ ($\overline{f}_k$) is a segment of consecutive words of document e (f). Then we compute the sentence similarity (translation quality) by iteration from the initial segment ($\overline{e}_1$, $\overline{f}_1$). If the similarity reaches the preset threshold for ($\overline{e}_i$, $\overline{f}_j$), we pick the segment pair as parallel sentence candidate, and continue the computation from ($\overline{e}_{i+1}$, $\overline{f}_{j+1}$).

We notice that the inconsistency of segmentation in the document pairs can lead to the results: a sentence in one language contains information only part of a sentence in the other language, or two sentences (in different languages) both contain part of their information in common. These resulting sentence pairs may have low similarity scores.

In order to alleviate this problem, we also incorporate a parallel segmentation method in our mining module. We follow the basic idea proposed in (Nevado et al., 2003) where the parallel segmentation finding problem is treated as an optimization problem and a dynamic programming scheme is

[1] https://github.com/moses-smt/mosesdecoder/blob/master/scripts/tokenizer/tokenizer.perl

[2] https://github.com/aboSamoor/polyglot

used to search for the best segmentation. We then briefly introduce our method. [3]

After obtaining the monolingual initial segmentation $(\overline{e}_1^a, \overline{f}_1^b)$, a parallel segmentation is represented as:

$$s \equiv ([\overline{e}_1^{k_1}, \overline{f}_1^{j_1}], [\overline{e}_{k_1+1}^{k_2}, \overline{f}_{j_1+1}^{j_2}], \cdots, [\overline{e}_{k_{|s|-1}+1}^{k_{|s|}}, \overline{f}_{j_{|s|-1}+1}^{j_{|s|}}]) \quad (3)$$

where $|s|$ is the number of segments for the parallel segmentation s and $\overline{e}_{i_1}^{i_2}$ ($\overline{f}_{i_1}^{i_2}$) are consecutive segments $\overline{e}_{i_1}\overline{e}_{i_1+1}\cdots\overline{e}_{i_2}$ ($\overline{f}_{i_1}\overline{f}_{i_1+1}\cdots\overline{f}_{i_2}$). In this setting, all initial segments will be included in the parallel segmentation, and the order of the initial segments cannot be inverse. Therefore, the alignment is monotone.

Then we search for the best parallel segmentation using the objective function:

$$\max_{S \in s} C \cdot \prod_{n=1}^{|s|} P(\overline{f}_{j_{n-1}+1}^{j_n} | \overline{e}_{k_{n-1}+1}^{k_n}) \quad (4)$$

where $P(\overline{f}_{j_{n-1}+1}^{j_n} | \overline{e}_{k_{n-1}+1}^{k_n})$ is the translation quality of the pair $(\overline{f}_{j_{n-1}+1}^{j_n}, \overline{e}_{k_{n-1}+1}^{k_n})$.

Next, we use a dynamic programming algorithm to compute the best segmentation S where we have a restriction that no more than 3 initial segments can be joined.

Finally, this set of parallel segments are combined with the set of the extracted initial segment pairs through global deduplication to serve as the output of our mining module.

It is worth noting that the sequence can be very long in the process of translation quality computation because several segments can be joined together. Therefore, while computing the similarity of a segment pair, our method based on word translation probability tables can be more time-efficient than LASER, as LASER is based on sentence embeddings and can be very slow when its LSTM encoder is fed with long sequence. Thus we do not consider using LASER to compute translation quality in our mining procedure.

2.2.3 Iterative Mining Strategy

The quantity and quality of the mined data are basically dependent on the word alignment model. Besides, more high-quality parallel corpus is used, the word alignment model would be more accurate and robust. Therefore, we propose an iterative mining strategy. For the first time, all the provided parallel data by the task are used to train the word alignment model. But we keep mining data for several times. We iteratively add new high-quality sentence pairs to the parallel corpus and train the word alignment model again to improve the word translation probability tables, thus boosting the mining cycle.

2.3 Scoring Module

The scoring module consists of three parts. First, we make use of both the parallel and monolingual data to train an XLM-based classifier to score each sentence pair. Secondly, different reranking mechanisms are used to adjust the scores. Finally, we ensemble four different models to improve the performance of our systems.

2.3.1 XLM-based Scorer

Recently, pre-trained transformer-based models play an important role in a variety of NLP tasks, such as question answering, relation extraction, etc.. Pre-trained models are often trained from scratch with self-supervised objective, and then fine-tuned to adapt to the downstream tasks. In our system, we choose the XLM (Conneau and Lample, 2019) as our main model. The reason are as follows: a) Similar to BERT (Devlin et al., 2019; Yang et al., 2019), XLM has **M**asked **L**anguage **M**odel (MLM) objective, which enables us to make the most use of the provided monolingual corpora; b) XLM also has **T**ranslation **L**anguage **M**odel (TLM) objective. Taking two parallel sentences as input, it predicts the randomly masked tokens. In this way, cross-lingual features can be captured; c) With a large amount of training corpus in different languages, XLM can provide powerful cross-lingual representation for downstream tasks, which is very suitable for parallel corpus filtering situations.

We follow the instructions [4] to prepare the training data and train the XLM model. In detail, we use Moses tokenizer to tokenize the text [5], and fastbpe [6] to learn and apply Byte-Pair Encoding. We use 50K BPE codes on the concatenation of all the training data. After applying BPE codes to the training data, we obtain a large vocabulary con-

[3] More details can be found in (Nevado et al., 2003)

[4] https://github.com/facebookresearch/XLM

[5] We do not use the character-/dictionary- based method introduced in Section 2.2.1 to tokenize Khmer here. Performance may be improved with that method, but we have run out of time, unfortunately.

[6] https://github.com/glample/fastBPE

taining around 100K tokens. Therefore, we only keep the top frequent $100,000$ tokens to form the vocabulary and train the XLM.

We use monolingual data in MLM objective and parallel data in TLM objective. In detail, the monolingual data we use are as follows:

- Khmer: all the 14M provided sentences.
- Pashto: all the 6.6M provided sentences.
- English: because the number of english monolingual sentences are so large, we subsample 25M sentences to keep a balance.

All the available parallel sentence pairs (29K en-km and 12K en-ps) are used in TLM objective. For each objective, we hold out 5K sentences or sentence pairs for validation and 5K for the test.

We pre-train the XLM using two different settings on 8 Tesla-V100 GPU: a) **Standard**: The embedding size is 1024, with 12 layers and 64 batch size. b) **Small**: The embedding size is 512, with 6 layers and 32 batch size. The other values of hyperparameters are all set to the default values. The two pre-trained XLM model is then fine-tuned in downstream task and further ensembled.

To score sentence pairs according to their parallelism, classification models are usually used (Xu and Koehn, 2017; Bernier-Colborne and Lo, 2019). In the training phrase, it is formulated as a binary classification problem, whether the sentence pair is semantically similar to each other or not. In the inference phrase, the probability of the positive class is considered as the score of the sentence pair. Therefore, we use the provided parallel sentence pairs as the positive instances, and construct negative instances taking advantage of such positive instances similar to Bernier-Colborne and Lo (2019). Specifically, we generate negative examples in the following ways:

- Shuffle the sentences in source language and target language respectively, and randomly align them.
- Randomly truncate the length of the source sentences or/and target sentences to 3.
- Randomly shuffle the order of the source sentences or/and target sentences.
- Simply swap the source and target sentences. Or replace the source/target sentences with target/source sentences, such that the two sentences are exactly the same.

We only add a linear or convolutional layer on top of the pre-trained XLM model and predict through a sigmoid function. The input of the model is the concatenation of a sentence pair, separated by one [SEP] token. Besides, to tackle the problem that some sentences may be too long, we simply truncate each sentence such that the maximum length of sentence is 128. The dropout rate is set to 0.5.

2.3.2 Reranking

We apply some reranking mechanisms in order to compensate for the latent bias in the XLM-based scorer, and aim to boost the quality of the whole corpus rather than each sentence pair independently.

The first reranking mechanism is based on **language identification**. For some sentences, they may include many tokens that do not belong to the corresponding language, and therefore damage the performance of the machine translation system. This phenomenon is rather common in Khmer-English corpus in particular. We utilize *pycld2* tools [7] to identify the language of the sentences. The scores of those which cannot be identified as the corresponding language are reranked by a discount of α. α is a hyperparameter.

The second reranking mechanism is based on **n-gram coverage**. Because the sentence pairs are scored independently, redundancy may exist in those high-score sentences. To enhance the diversity of the selected corpus, we first sort the sentence pairs in the descending order based on their scores. Next we maintain a n-gram pool for source sentences, and scan the source sentences from the top down. Those sentences that have no n-gram different from those in the pool will receive a discount of β, and both n and β are hyperparameters.

Note that before reranking, we always normalize the score according to their rankings, so that scores provided by different models can be unified. The score of the i-th sentence pair is:

$$score_i = 1 - \frac{rank_i}{N} \tag{5}$$

where i-th pair ranks $rank_i$ in all the sentence pairs and N denotes total number of pairs.

We also try to rerank through language models, but it does not bring improvements. Thus we do not use this reranking mechanism in our submissions.

[7] https://github.com/aboSamoor/pycld2

2.3.3 Ensemble

Different models may capture different features during training and inference. To make use of group wisdom and improve the final performance, we ensemble the following four models by averaging scores:

- Model 1: Standard XLM + Linear Layer. The learning rate of XLM and linear layer are $1e^{-8}$ and $1e^{-5}$ respectively.
- Model 2: Standard XLM + Linear Layer. The learning rate of XLM and linear layer are $1e^{-7}$ and $1e^{-4}$ respectively.
- Model 3: Standard XLM + Convolutional Layer. The learning rate of XLM and linear layer are $5e^{-7}$ and $5e^{-4}$ respectively.
- Model 4: Small XLM + Linear Layer. The learning rate of XLM and linear layer are $5e^{-7}$ and $5e^{-4}$ respectively.

All the models use 16 batch size per GPU.

3 Experiments

We conduct various experiments to evaluate the performance of different models, and select the most proper hyperparameters for both Khmer-English and Pashto-English. Note that **FS** and **FT** denote **From Sratch** and **Fine-Tune** respectively.

Firstly, we conduct the experiments with both the provided aligned sentence pairs (denoted as **Baseline**) and our mined data at the first iteration of the mining module. It shows that our system can outperform the baseline remarkably and the ensemble of four different models can further improve the performance. As Table 2 illustrates, Model 1-4 outperform baseline by about $1 \sim 2$ BLEU in both km-en and ps-en . Besides, the ensemble model performs the best in general.

Table 2: BLEU Scores of Difference Models

Model	km-en		ps-en	
	FS	FT	FS	FT
Baseline	7.28	10.24	9.81	11.37
Model 1	8.33	11.43	11.21	13.11
Model 2	8.96	11.38	**11.43**	13.18
Model 3	8.72	11.29	11.26	12.74
Model 4	9.01	11.27	11.36	13.09
Ensemble	**9.22**	**11.51**	11.28	**13.52**

Next, to verify the effectiveness of the iterative mining strategy in the mining module, we compare the performance of the same ensemble model with different mined data. In our paper, we iteratively mine data for three times, and combine them with the provided sentence-aligned corpus. Table 3 reveals the mining scale each time. As table 4 shows, iteration 3 works best for km-en and iteration 2 for ps-en respectively.

Table 3: The Number of Mined Sentence Pairs

Data	km-en	ps-en
Data 1	238K	200K
Data 2	330K	120K
Data 3	660K	20K

Table 4: BLEU Scores with Different Mined Data

Data	km-en		ps-en	
	FS	FT	FS	FT
+ Data 1	9.22	11.51	11.28	13.52
+ Data 1+2	9.47	11.56	**12.17**	**13.19**
+ Data 1+2+3	**9.84**	**11.62**	12.14	12.69

Finally, by introducing the reranking mechanism, we can further improve the performance, which is shown by Table 5 and 6. Note that $\alpha = 0$ or $\beta = 0$ means it does not have any discount. We select $\alpha = 0.2, n = 2, \beta = 0.2$ and $\alpha = 0, n = 1, \beta = 0.1$ for km-en and ps-en for our submissions.

Table 5: BLEU Scores with Reranking for km-en

	FS	FT
$\alpha = 0, \beta = 0$	9.84	11.62
$\alpha = 0.2, n = 2, \beta = 0.05$	10.40	12.25
$\alpha = 0.2, n = 2, \beta = 0.1$	10.38	11.87
$\alpha = 0.2, n = 2, \beta = 0.2$	**10.50**	**12.45**
$\alpha = 0.2, n = 3, \beta = 0.1$	10.09	12.09
$\alpha = 0.3, n = 2, \beta = 0.05$	10.40	12.25

4 Conclusion

In this paper, we present our submissions to the WMT20 shared task on parallel Corpus filtering and alignment for low-resource conditions. Our Volctrans system consists of two modules: a) Mining module is responsible for mining potential parallel sentence pairs out of the provided document pairs. Word alignment model is utilized and an iterative mining strategy is further taken to boost

Table 6: BLEU Scores with Reranking for ps-en

	FS	FT
$\alpha = 0, \beta = 0$	12.17	13.19
$\alpha = 0, n = 1, \beta = 0.1$	**12.28**	13.34
$\alpha = 0.2, n = 1, \beta = 0.1$	12.15	13.06
$\alpha = 0, n = 2, \beta = 0.1$	12.20	**13.38**
$\alpha = 0, n = 2, \beta = 0.2$	12.20	13.31

the mining performance. b) Scoring module aims to evaluate sentence pairs quality according to their parallelism and fluency properties, by exploiting an XLM-based scorer. We further tune the output score with different reranking mechanism, by considering language detection confidence and n-gram vocabulary coverage. Finally, four models are ensembled to improve the final performance. We also make some analysis through a variety of experiments.

References

2008. *Proceedings of the International Conference on Language Resources and Evaluation, LREC 2008, 26 May - 1 June 2008, Marrakech, Morocco*. European Language Resources Association.

Mikel Artetxe and Holger Schwenk. 2019. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Trans. Assoc. Comput. Linguistics*, 7:597–610.

Gabriel Bernier-Colborne and Chi-kiu Lo. 2019. NRC parallel corpus filtering system for WMT 2019. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 252–260, Florence, Italy. Association for Computational Linguistics.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 7057–7067.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648, Atlanta, Georgia. Association for Computational Linguistics.

Philipp Koehn, Francisco Guzmán, Vishrav Chaudhary, and Juan Pino. 2019. Findings of the WMT 2019 shared task on parallel corpus filtering for low-resource conditions. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 54–72, Florence, Italy. Association for Computational Linguistics.

Chi-kiu Lo. 2019. YiSi - a unified semantic MT quality evaluation and estimation metric for languages with different levels of available resources. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 507–513, Florence, Italy. Association for Computational Linguistics.

Francisco Nevado, Francisco Casacuberta, and Enrique Vidal. 2003. Parallel corpora segmentation using anchor words. In *Proceedings of the 7th International EAMT workshop on MT and other language technology tools, Improving MT through other language technology tools, Resource and tools for building MT at EACL 2003*.

Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. 2018. Scaling neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 1–9, Belgium, Brussels. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 5998–6008.

Hainan Xu and Philipp Koehn. 2017. Zipporah: a fast and scalable data cleaning system for noisy web-crawled parallel corpora. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2945–2950, Copenhagen, Denmark. Association for Computational Linguistics.

Jiacheng Yang, Mingxuan Wang, Hao Zhou, Chengqi Zhao, Yong Yu, Weinan Zhang, and Lei Li. 2019. Towards making the most of bert in neural machine translation. *arXiv preprint arXiv:1908.05672*.

Jinhua Zhu, Yingce Xia, Lijun Wu, Di He, Tao Qin, Wengang Zhou, Houqiang Li, and Tie-Yan Liu. 2020. Incorporating BERT into neural machine translation. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*.

PATQUEST: Papago Translation Quality Estimation

Yujin Baek[*,†]
Graduate School of AI, KAIST
yujinbaek@kaist.ac.kr

Zae Myung Kim[*]
Papago, Naver Corp.
zaemyung.kim@navercorp.com

Jihyung Moon
Papago, Naver Corp.
jihyung.moon@navercorp.com

Hyunjoong Kim
Papago, Naver Corp.
soy.lovit@navercorp.com

Eunjeong L. Park
Papago, Naver Corp.
lucy.park@navercorp.com

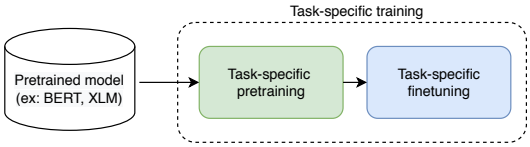
Abstract

This paper describes the system submitted by Papago team for the quality estimation task at WMT 2020. It proposes two key strategies for quality estimation: (1) task-specific pretraining scheme, and (2) task-specific data augmentation. The former focuses on devising learning signals for pretraining that are closely related to the downstream task. We also present data augmentation techniques that simulate the varying levels of errors that the downstream dataset may contain. Thus, our PATQUEST models are exposed to erroneous translations in both stages of task-specific pretraining and finetuning, effectively enhancing their generalization capability. Our submitted models achieve significant improvement over the baselines for Task 1 (Sentence-Level Direct Assessment; EN-DE only), and Task 3 (Document-Level Score).

1 Introduction

With the widespread use of machine translation systems, there is a growing need to evaluate translated results at low-cost. The task of quality estimation (QE) addresses this issue, where the quality of a translation is predicted automatically given the source sentence and its translation. The estimated quality can inform users about the reliability of the translation, or whether it needs to be post-edited.

Previous QE systems generally include pretraining and finetuning steps, where the former step involves masked language modeling (MLM) utilizing large parallel corpora, with the expectation that the models will learn cross-lingual relationships (Kepler et al., 2019; Kim et al., 2019). The models are, in turn, finetuned with task-specific data. However, while the pretraining step involves training data with near-perfect translations, low-quality translations are only introduced during the finetuning step.



	Task1 (Section 2)	Task3 (Section 3)
Perturbations	- Omitted word - Word order - Lexical selection - Repeated phrase	- Omitted word (preposition, determiner) - Word order - Lexical selection - Wrong preposition
Label	- SentenceBERT - BERTScore - GPT-2	- Pseudo-MQM

Figure 1: Overview of our approach for Task 1 and 3.

In this work, we suggest two key strategies that could alleviate this pretrain-finetune discrepancy in QE tasks by: (1) adopting a task-specific pretraining objective which is close to that of the downstream task, and (2) generating abundant task-specific erroneous sentence pairs and their learning signals. Our approach, which is depicted in Figure 1, is motivated from BLEURT (Sellam et al., 2020), where we extend their general approach to the bilingual QE setting. Our submitted systems achieve significant improvements in performance over the baseline systems on WMT20 Shared Tasks for QE (Specia et al., 2020): an absolute gain of $+35.2\%$ in Pearson score for (Task 1) Sentence-Level Direct Assessment (EN-DE), and $+18.4\%$ in Pearson score for (Task 3) Document-Level Score.

2 Sentence-Level QE: Direct Assessment

The task of sentence-level QE for direct assessment (DA) involves predicting the perceived quality of the translation given the source and the translated sentences.

Following the footsteps of the previous work

[*] Equal contribution
[†] Work done during internship at Naver Corp.

Proceedings of the 5th Conference on Machine Translation (WMT), pages 991–998
Online, November 19–20, 2020.

on QE, our sentence-level system also utilizes the pretrained multilingual language models such as BERT (Devlin et al., 2018) and Cross-lingual Language Model (XLM) (Conneau and Lample, 2019). As the size of the training corpus for the QE task is very limited (7K sentence pairs), it is crucial to align these models closely to the task using more data in the form of task-specific pretraining.

As opposed to pretraining the models on parallel corpora using the standard MLM approach, we pretrain the models in a multi-task setting using learning signals and data that are arguably more task-specific similar to Sellam et al. (2020).

2.1 Task-Specific Data Augmentation

In order to better align the pretrained models to the QE task, synthetic sentence pairs that contain various types of translation errors are generated from clean parallel corpora[1]. For each target sentence, we generate two perturbed sentences by separately applying one of the four methods described below.

Omitted Word We randomly omit at most three words from the target-side, simulating inadequate translations.

Word Order Based on the part-of-speech (POS) tag for each word in the target sentence, and predefined sequences of POS patterns, we randomly swap two target words if those words match one of the patterns. The POS patterns can be contiguous, e.g., *adjective-space-noun*, or long-ranged, e.g., *noun-*-adjective*. When none of the patterns are matched, we randomly swap two words.

Lexical Selection For each target sentence, we mask out at most three words randomly, and apply mask-filling via a German BERT model from Hugging Face[2]. The purpose of this alteration is to generate fluent but somewhat inadequate target sentences.

Repeated Phrase In order to simulate the repetition problem in translations generated by neural machine translation models, we alter the target sentence by adding a repetition of a random phrase within the sentence. The length of the random phrase is at most three tokens.

[1] `Europarl v10` and `News Commentary v15`

[2] `bert-base-german-cased`, https://huggingface.co/transformers/pretrained_models.html

2.2 Task-Specific Learning Signals

As the goal of the downstream task is to predict the DA scores which represent the "perceived quality" of the translation, we need to consider pretraining signals that can capture the somewhat subjective notion of "good" and "bad" translations.

Consulting the related works, we prepared the three learning signals:

- SentenceBERT score (Reimers and Gurevych, 2019)
- BERTScore (Zhang et al., 2019), extended to multilingual setting
- Target (German) Language Model (GPT-2, Radford et al. (2019)) score

For each sentence pair in the original bilingual corpora as well as the augmented ones, the three types of learning signals are computed, and later used in the task-specific pretraining.

2.2.1 SentenceBERT Score

For a given sentence, SentenceBERT produces a semantically meaningful sentence embedding that can be compared using a distance metric.

We note that when comparing the distance between two sentence vectors, the Kendall rank correlation coefficient (Kendall, 1938) is computed instead of the cosine similarity measure as the former correlates better with the human judgement, possibly because it produces a more widespread range of scores than the latter especially when the dimension of the sentence vectors is high.

In our experiments, we used the publicly available multilingual SentenceBERT model released from UKPLab[3] that supports 13 languages including English and German.

2.2.2 Multilingual BERTScore

While SentenceBERT score looks at the sentence embedding as a whole, BERTScore computes a similarity score for each token in the pair of sentences. We include BERTScore as one of the learning signals because we feared that the mean-pooling of the BERT-embedded tokens within the SentenceBERT model, while effective in extracting the overall meaning of the sentence, may overlook some of the small semantic details within the sentence.

[3] `distiluse-base-multilingual-cased`, https://github.com/UKPLab/sentence-transformers

However, as the original BERTScore is designed to work in monolingual setting, i.e. evaluating a translation against a reference sentence, it needs to be extended in multilingual setting using a multilingual BERT (mBERT) model. Analogous to the original approach, the multilingual BERTScores can be computed in various ways depending on which side we are computing the maximum similarities from.

In our experiments, we devise a metric where we merge both the source- and target-side maximum similarities between tokens with the corresponding inverse document frequency (IDF) weighting; thus, given a sequence of vectorized source and target tokens, s and t, we defined the `mBERTScore` of s and t to be:

$$\frac{S_{s\to t} + S_{t\to s}}{\sum_{s_i \in s} \text{idf}(s_i) + \sum_{t_j \in t} \text{idf}(t_j)}$$

where

$$S_{s\to t} = \sum_{s_i \in s} \text{idf}(s_i) \max_{t_j \in t} \boldsymbol{s_i}^\top \boldsymbol{t_j}$$

$$S_{t\to s} = \sum_{t_j \in t} \text{idf}(t_j) \max_{s_i \in s} \boldsymbol{t_j}^\top \boldsymbol{s_i}$$

2.2.3 Target Language Model Score

While SentenceBERT and multilingual BERTScore can be used as proxies for evaluating the "adequacy" of the translation, empirically, we noticed that they cannot seem to sufficiently represent the "fluency" of translated target sentence. In other words, both metrics may assign high scores to the translated sentence if key source tokens are translated and present in the translation, even when the overall sentence may not be articulate.

To address this issue, the target language model (GPT-2) score is added to the set of learning signals. We simply use the arithmetic mean of the token-level predictions to produce the score for a target sentence. We utilize the pretrained GPT-2 model for German released by Zamia Brain[4].

2.3 Model Architecture

We have two stages for task-specific training, i.e. first with the augmented data and the learning signals, and second with the provided QE dataset (ref. Section 2.4). As the output to predict for each stage is different, we utilize the following two types of model architectures.

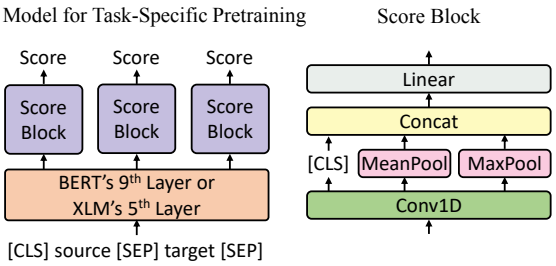


Figure 2: The model architecture (*left*) for the task-specific pretraining using the augmented dataset and learning signals. It consists of three separate Score Blocks (*right*) added on top of the BERT's or XLM's layer.

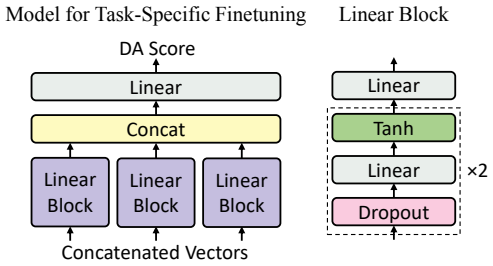


Figure 3: The model architecture (*left*) for the task-specific finetuning using the provided QE dataset. For each concatenated vector computed within each Score Block (c.f. Fig. 2.), a Linear Block (*right*) is added on top of it. The results from the Linear Blocks are concatenated and used to produce the final DA score.

2.3.1 Model for Task-Specific Pretraining

On top of the specific layer of the pretrained mBERT or XLM models, we attach a series of layers called "Score Block" for each type of learning signal as depicted in Figure 2. We utilize the 9th and 5th layer of the BERT and XLM models, respectively, as these layers are reported to be more semantically relevant (Jawahar et al., 2019; Zhang et al., 2019).

In addition to using the vector representation of the `[CLS]` token, utilizing the mean-pooled and max-pooled vectors from all tokens further improved the performance.

2.3.2 Model for Task-Specific Finetuning

Once the task-specific pretraining is completed, we begin the finetuning by adding layers above the concatenation layer within each Score Block, as

[4] `gpt2-german-345M-r20191119`, http://zamia-speech.org/brain

shown in Figure 3. Thus, we have three concatenated vectors being fed to three "Linear Blocks" separately, whose purpose is to reduce the dimensions of the hidden representation, preparing it for the final regression layer.

We note that applying dropout (Srivastava et al., 2014) to these linear layers helps with the performance.

2.4 Task-Specific Training

We experiment with three different types of pretrained models: mBERT[5], XLM trained with MLM (XLM-MLM)[6], and XLM trained with causal language modeling (XLM-CLM)[7]. All of the pretrained models are available at Hugging Face.

2.4.1 Task-Specific Pretraining (TSP)

As the size of the provided QE dataset is small, we make use of the existing parallel data as well as the error-induced synthetic data. For the EN-DE bilingual dataset, we select a subset from this year's training corpora for WMT News Translation Task, summing to just under 10M sentence pairs; for the synthetic dataset, the size is 3.4M.

Given the concatenated source and target sentences as an input, the model for TSP is trained to predict the three types of learning signals in a multi-task setting by minimizing the sum of the mean squared error losses for each signal (ref. Figure 2).

2.4.2 Task-Specific Finetuning (TSF)

Once the model is trained with the augmented data, its parameters are loaded to the model for TSF (ref. Figure 3), and finetuned using the QE dataset. This time, the model learns to predict the mean z-normalized DA score.

3 Document-Level QE: MQM Scoring

Given a source and its translated document, this task involves identifying translation errors and estimating the translation quality of the document based on the taxonomy of the Multidimensional Quality Metrics (MQM)[8]. With the pre-defined MQM taxonomy, human annotators assess whether the translation satisfies the specifications, and from these annotations, an MQM score is obtained. In this work, we focus on building a system that predicts the MQM score for a given pair of source and translated document.

The major difficulty that we encountered in this task was the lack of training data. As the amount of provided data is limited (8,591 sentence pairs), a model that is solely finetuned on this small-scale data was not capable enough to differentiate sentences with varying level of errors.

To address this issue, we propose simple yet effective methods for task-specific data augmentation, and task-specific training framework[9].

3.1 Task-Specific Data Augmentation

We generate erroneous sentence pairs and their pseudo-MQM scores from Europarl and QE training corpus in accordance with the MQM taxonomy.

3.1.1 Generating Erroneous Sentence Pairs

Out of the 45 error categories specified in QE annotations, we select five frequent categories for which we can automatically perturb the target-side of the parallel corpus at little cost. More details on our data augmentation technique for each category are provided below.

Omitted Preposition We introduce an error into the target-side of a sentence pair by randomly omitting one of the French prepositions that exist in the sentence.

Omitted Determiner The same process is done for French determiners as for prepositions.

Wrong Preposition We replace a French preposition with another one. When more than one candidate exists, we choose one at random.

Word Order We exploit grammatical pattern that most descriptive adjectives go after the noun in French sentences (unlike English ones). Using an in-house French POS tagger, we identify post-nominal adjectives and place them in front of the corresponding nouns so that they are now pre-nominal.

Lexical Selection We mask-out target tokens at random positions, and substitute them with tokens predicted by the Camembert language model (Martin et al., 2020).

[5] bert-base-multilingual-cased
[6] xlm-mlm-ende-1024
[7] xlm-clm-ende-1024
[8] http://www.qt21.eu/mqm-definition

[9] The code will be available at https://github.com/naver/PATQUEST.

Error name	Sentence	Length	Total error severity	Pseudo MQM
Original sentence	Vous avez souhaité un débat à ce sujet dans les prochains jours, au cours de cette période de session.	21	0	100.0
(1) Wrong Preposition	Vous avez souhaité un débat à ce sujet *chez* les prochains jours, au cours de cette période de session.	21	5	76.2
(2) Omit Determiner	Vous avez souhaité ~~un~~ débat à ce sujet dans les prochains jours, au cours de cette période de session.	21	5	76.2
(1)+(2)	Vous avez souhaité ~~un~~ débat à ce sujet *chez* les prochains jours, au cours de cette période de session.	20	10	52.4
Original sentence	Cela placerait l'UE dans une situation délicate vis-à-vis de ces pays et de la communauté internationale.	23	0	100.0
(1) Word Order	Cela placerait l'UE dans une situation délicate vis-à-vis de ces pays et de la *internationale communauté*.	23	5	78.3
(2) Lexical Selection	Cela placerait l'UE dans une situation *inconfortable* vis-à-vis de ces pays et de la communauté internationale.	23	5	78.3
(1)+(2)	Cela placerait l'UE dans une situation *inconfortable* vis-à-vis de ces pays et de la *internationale communauté*.	23	10	56.5

Table 1: Examples of erroneous sentence pairs generated from the Europarl corpus.

Error name	Sentence	Length	Total error severity	Pseudo MQM
Original sentence	son travail a été présenté dans le washington post, *quotidien bonbons*, washingtonian, fit yoga et journal *d*'yoga.	23	15	34.8
(1) Wrong Preposition	son travail a été présenté *pour* le washington post, *quotidien bonbons*, washingtonian, fit yoga et journal *d*'yoga.	23	20	13.0
(2) Omit Determiner	son travail a été présenté dans ~~le~~ washington post, *quotidien bonbons*, washingtonian, fit yoga et journal *d*'yoga.	22	20	9.1
(1)+(2)	son travail a été présenté *pour* ~~le~~ washington post, *quotidien bonbons*, washingtonian, fit yoga et journal *d*'yoga.	22	25	-13.6
Original sentence	Brûleur deux *plaque* de cuisson anti-adhésive de Coghlan	10	5	50.0
(1) Omit Preposition	Brûleur deux *plaque* de cuisson anti-adhésive ~~de~~ Coghlan	9	10	-11.1

Table 2: Examples of erroneous sentence pairs generated from the WMT20 QE corpus.

3.1.2 Task-Specific Learning Signal

Once we introduce different types of errors into the target-side sentences, the next step is to obtain pseudo-MQM scores for the altered sentence pairs. Two key elements for computing MQM score are the length of a text, and its total error severity as follows:

$$\text{Pseudo-MQM} = 100(1 - \frac{5.0 * n_{error} + S}{N})$$

where N indicates the length of given target sentence and n_{error} denotes the number of errors introduced in it. We assign 5.0, the most frequent severity, to each perturbation that we make. If an error severity score, S, is assigned to the sentence by human annotators, we add this score to compute the total error severity score.

3.2 Model Architecture

We use pretrained mBERT or XLM[10] as initial parameters. The concatenation of a source sentence and its corresponding target sentence with special symbol tokens is taken as input: `[CLS] source [SEP] target [SEP]`.

We experiment with two strategies for obtaining sentence embeddings. First, we feed a hidden state vector corresponding to `[CLS]` token ($h_{[CLS]}$) to a linear layer to compute a sentence-level MQM prediction of $\hat{y}$:

$$\hat{y} = W h_{[CLS]} + b$$

where W and b are the weight matrix and bias vector of the linear layer, respectively. For the other method, we use the concatenation of a mean-pooled source representation ($s \in \mathbb{R}^n$), mean-pooled target representation ($t \in \mathbb{R}^n$) and their element-wise differences ($|s - t| \in \mathbb{R}^n$) in an attempt to enlarge the model capacity:

$$\hat{y} = W \cdot \text{ReLU}(W_r(s, t, |s - t|) + b_r) + b$$

where $W_r \in \mathbb{R}^{3n \times n}$ and b_r are the weight matrix and bias vector of an intermediate dimension-reducing layer, respectively, and n denotes the dimension of hidden vectors. W and b are the weight matrix and bias vector of the final linear layer.

3.3 Task-Specific Training

We suggest that the pretraining objective should be similar to that of the downstream task in order to mitigate the pretrain-finetune discrepancy (Yang et al., 2019), and fully leverage the erroneous sentence pairs that we generated. For this task, both phases minimize the mean-squared loss function: $l = \frac{1}{K} \sum_{k=1}^{K} \|y_k - \hat{y}\|^2$.

3.3.1 Task-Specific Pretraining (TSP)

We utilize Europarl parallel corpus (English-French) to pretrain our submitted models[11]. To acquire high quality data, we carried out the following filtering processes: (1) language detection (filtering out non-English sentences in the source-side, and non-French sentences in the target-side), (2) length ratio filtering (eliminating sentence pairs with length ratio greater than 1.8).

[10] `xlm-mlm-enfr-1024`

[11] We perform TSP after bringing pretrained parameters of language models as initial weights.

We assume that the remaining sentence pairs do not contain any translation error. Therefore, we assign the total error severity score of zero to these pairs before the augmentation.

About 15.2 million examples[12] are generated with the above-mentioned data augmentation techniques. The detailed examples are provided in Table 1.

3.3.2 Task-Specific Finetuning (TSF)

The next step is to finetune our model using the augmented QE train data. Unlike Europarl corpus, we can fully leverage the MQM scores originally assigned to the QE training dataset. We found that performing the data augmentation with three categories (*Omitted Determiner*, *Omitted Preposition*, and *Wrong Preposition*) effectively improves the performance. The original QE training sentence pairs represent about 5% of 169,997 sentence pairs obtained from the data augmentation. We also provide the augmented examples for QE training data in Table 2.

Since the learning objective is identical to that of the pretraining phase, we can simply train the same model with the augmented downstream task data.

3.4 Document-Level MQM Score

We specify that the models are trained at sentence-level, learning to predict the non-truncated version of MQM scores which could take a range between negative infinity and 100; this is to avoid potential information loss that could arise from the truncation.

Given a document, the document-level MQM score is computed from its sentence-level MQM predictions in a closed form. Afterwards, we truncate negative values to zero.

4 Experimental Results

4.1 Sentence-Level Task

Table 3 shows the Pearson correlation coefficient between the predicted z-normalized DA scores and the reference scores on the development set. We note that the number of parameters for PATQUEST-mBERT (724M) is greater than that of PATQUEST-XLM (616M) models, resulting in the difference in the correlation scores. Nevertheless, computing the arithmetic mean of the scores produced by these three models improves the performance (PATQUEST-ensemble).

Model	Pearson's r ↑
PATQUEST-mBERT	0.486
PATQUEST-XLM-MLM	0.450
PATQUEST-XLM-CLM	0.452
PATQUEST-ensemble	**0.501**

Table 3: Results on the *development* set for Task 1 EN-DE.

Model	Pearson's r ↑	MAE ↓	RMSE ↓
Baseline	0.146	0.679	0.967
PATQUEST-mBERT w/o synth. data	0.429	0.462	0.632
PATQUEST-ensemble w/o synth. data	0.457	0.464	0.640
PATQUEST-ensemble	**0.498**	**0.454**	**0.637**

Table 4: Submission results on the *test* set for Task 1 EN-DE.

The final result on the QE test set is shown in Table 4. We observe that finetuning the model with the additional error-induced synthetic data improves the performance as well as ensembling the models.

Our final submitted system (PATQUEST-ensemble) finished 4th out of the 15 submitted systems[13] in the final ranking of the sentence-level QE task for English-German. In order to train a generally applicable QE system, we did not make use of the data such as internal information from the NMT models and in-domain Wikipedia texts that could be extracted from the provided Wikipedia titles.

4.2 Document-Level Task

The validation results on development set are shown in Table 5. Both PATQUEST-mBERT and PATQUEST-XLM models use representations from [CLS] token. We build another two models, PATQUEST-mBERT variant 1 and 2, using the concatenations of mean-pooled source representations, mean-pooled target representations, and their element-wise differences.

Table 6 shows the test results of our submitted PATQUEST models. For PATQUEST-ensemble, we compute an average from the four models enumerated in Table 5.

In Table 7, the effectiveness of our training scheme and data augmentation techniques is illustrated via an ablation study. Note that "Pretrained mBERT (A)" in the table refers to the mBERT

[12]The size of the original Europarl English-French parallel corpus is about 2M sentence pairs.

[13]Excluding the disqualified team.

Model	Pearson's r ↑	MAE ↓	RMSE ↓
PATQUEST-mBERT	0.431	14.401	22.330
PATQUEST-mBERT variant 1	0.406	14.418	22.872
PATQUEST-mBERT variant 2	0.380	14.909	23.215
PATQUEST-XLM	0.374	16.245	23.647

Table 5: Results on the *development* set of WMT20 document-level task.

Model	Pearson's r ↑	MAE ↓	RMSE ↓
Baseline	0.389	19.939	26.608
PATQUEST-mBERT	0.529	16.214	24.437
PATQUEST-XLM	0.546	15.821	23.846
PATQUEST-ensemble	**0.573**	**15.611**	**23.327**

Table 6: Submission results of PATQUEST models on the *test* set of WMT20 document-level task.

model that is finetuned on the original QE data without any task-specific training. Both TSP and TSF enhance the generalization ability of model. Note that the mBERT model trained via TSP and TSF, "A + TSP + TSF", is the same model as PATQUEST-mBERT which itself achieves a significant improvement over the baselines as shown in Table 6.

Our final system (PATQUEST-ensemble) submitted for the document-level QE task, came 1st out of the three submitted systems[14]. Similar to our sentence-level system, our document-level system also did not utilize any internal information from the NMT models and in-domain Wikipedia data tailored to the benchmark.

5 Conclusion

In this paper, we present a task-specific pretraining scheme for the QE task. Our pretraining objective is devised so that it is closely related (Task 1) or identical (Task 3) to the finetuning objective. In addition, the models are exposed to abundant amount of error-induced translations generated from large parallel corpora, effectively alleviating the issue of data scarcity.

Our proposed models yield significant improvement over the baseline systems for the two tasks.

[14]Excluding the disqualified team

Model	Pearson's r ↑	MAE ↓	RMSE ↓
Pretrained mBERT (A)	0.263	16.146	23.090
A + TSF	0.341 (+ 0.078)	15.302	23.749
A + TSP	0.375 (+ 0.112)	15.496	23.444
A + TSP + TSF	0.431 (+ 0.168)	14.401	22.330

Table 7: Results on the *development* set of WMT20 document-level task adding up key components of our model.

Acknowledgments

Authors would like to thank Stéphane Clinchant, Vassilina Nikoulina, and Jaesong Lee for the insightful discussions, and Papago team members for offering the fruitful feedback. We would also like to extend our gratitude to Won Ik Cho for coming up with the awesome name for our system.

References

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems*, pages 7059–7069.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

Maurice G Kendall. 1938. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93.

Fabio Kepler, Jonay Trénous, Marcos Treviso, Miguel Vera, António Góis, M Amin Farajian, António V Lopes, and André FT Martins. 2019. Unbabel's participation in the wmt19 translation quality estimation shared task. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 78–84.

Hyun Kim, Joon-Ho Lim, Hyun-Ki Kim, and Seung-Hoon Na. 2019. Qe bert: Bilingual bert using multi-task learning for neural quality estimation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 85–89.

Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric Villemonte de la Clergerie, Djamé Seddah, and Benoît Sagot. 2020. Camembert: a tasty french language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Thibault Sellam, Dipanjan Das, and Ankur P Parikh. 2020. Bleurt: Learning robust metrics for text generation. *arXiv preprint arXiv:2004.04696*.

Lucia Specia, Frédéric Blain, Marina Fomicheva, Erick Fonseca, Vishrav Chaudhary, Francisco Guzmán, and André FT Martins. 2020. Findings of the wmt 2020 shared task on quality estimation. In *Proceedings of the Fifth Conference on Machine Translation: Shared Task Papers*.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5753–5763.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

RTM Ensemble Learning Results at Quality Estimation Task

Ergun Biçici
ergun.bicici@boun.edu.tr
Electrical and Electronics Engineering Department, Boğaziçi University
orcid.org/0000-0002-2293-2031

Abstract

We obtain new results using referential translation machines (RTMs) with predictions mixed and stacked to obtain a better mixture of experts prediction. We are able to achieve better results than the baseline model in Task 1 subtasks. Our stacking results significantly improve the results on the training sets but decrease the test set results. RTMs can achieve to become the 5th among 13 models in ru-en subtask and 5th in the multilingual track of sentence-level Task 1 based on MAE.

1 Introduction

Quality estimation task in WMT20 (Specia et al., 2020) (QET20) address machine translation (MT) performance prediction (MTPP), where translation quality is predicted without using reference translations, at the sentence- (Tasks 1 and 2), word- (Task 2), and document-levels (Task 3). Task 1 predicts the sentence-level direct assessment (DA) in 7 language pairs categorized according to the MT resources available:

- high-resource, English–German (en-de), English–Chinese (en-zh), and Russian-English (en-ru)
- medium-resource, Romanian–English (ro-en) and Estonian–English (et-en), and
- low-resource, Sinhalese–English (si-en) and Nepalese–English (ne-en).

en-ru contains sentences from both Wikipedia and Reddit articles while others use only Wikipedia sentences with 7000 sentences for training, 1000 for development, and 1000 for testing. The target to predict in Task 1 is z-standardised DA scores, which changes the range from $[0, 100]$ for DA scores to $[3.178, -7.542]$ in z-standardized DA scores.

			RTM interpretants		
Task	Train	Test	setting	Training	LM
Task 1 (en-de)	8000	1000	bilingual	0.3 M	5 M
Task 1 (en-zh)	8000	1000	monolingual en	0.2 M	3.5 M
Task 1 (si-en)	8000	1000	monolingual en	0.2 M	3.5 M
Task 1 (ne-en)	8000	1000	monolingual en	0.2 M	3.5 M
Task 1 (et-en)	8000	1000	monolingual en	0.2 M	3.5 M
Task 1 (ro-en)	8000	1000	monolingual en	0.2 M	3.5 M
Task 1 (ru-en)	8000	1000	bilingual	0.2 M	4 M
Task 2 (en-de)	8000	1000	bilingual	0.3 M	5 M
Task 2 (en-zh)	8000	1000	monolingual en	0.2 M	3.5 M

Table 1: Number of instances in the tasks and the size of the interpretants used.

The target to predict in Task 2 is sentence HTER (human-targeted translation edit rate) scores (Snover et al., 2006) and binary classification of word-level translation errors. We participated in sentence-level subtasks, which include English-German and English-Chinese in Task 2. Table 1 lists the number of sentences in the training and test sets for each task and the number of instances used as interpretants in the referential translation machine (RTM) (Biçici, 2018; Biçici and Way, 2015) models (M for million).

We tokenize and truecase all of the corpora using Moses' (Koehn et al., 2007) processing tools.[1] LMs are built using `kenlm` (Heafield et al., 2013).

2 RTM for MTPP

We use RTM models for building our prediction models. RTMs predict data translation between the instances in the training set and the test set using interpretants, data selected close to the task instances in bilingual training settings or monolingual language model (LM) settings. Interpretants provide context for the prediction task and are used during the derivation of the features measuring the closeness of the test sentences to the

[1] https://github.com/moses-smt/mosesdecoder/tree/master/scripts

Proceedings of the 5th Conference on Machine Translation (WMT), pages 999–1003
Online, November 19–20, 2020.

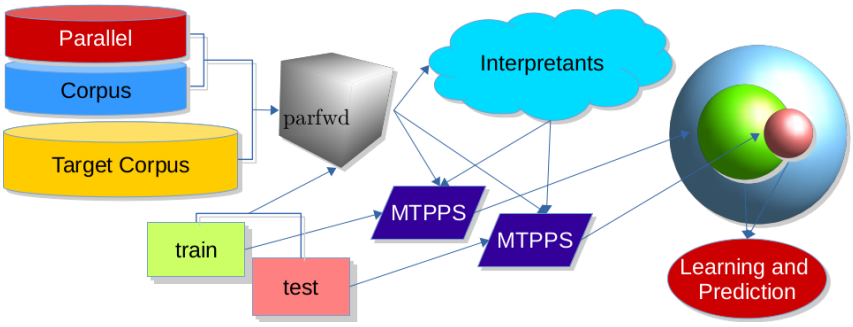


Figure 1: RTM depiction: parfwd selects interpretants close to the training and test data using parallel corpus in bilingual settings and monolingual corpus in the target language or just the monolingual target corpus in monolingual settings; an MTPPS use interpretants and training data to generate training features and another use interpretants and test data to generate test features in the same feature space; learning and prediction takes place using these features as input.

training data, the difficulty of translating them, and to identify translation acts between any two data sets for building prediction models. With the enlarging parallel and monolingual corpora made available by WMT, the capability of the interpretant datasets selected to provide context for the training and test sets improve as can be seen in the data statistics of `parfwd` instance selection, parallel feature weight decay (Biçici, 2019). RTMs use `parfwd` for instance selection and machine translation performance prediction system (MTPPS) (Biçici et al., 2013; Biçici and Way, 2015) for obtaining the features, which includes additional features from word alignment. Figure 1 depicts RTMs and explains the model building process.

Additionally, we included the sum, mean, standard deviation, minimum, and maximum of alignment word log probabilities as features in Task 1. In Task 2, we included word alignment displacement features including the average of source and target displacements relative to the length of the source or target sentences respectively and absolute displacement relative to the maximum of source and target sentence lengths.

Instead of resource based discernment, we treated en-de of Tasks 1 and 2 and ru-en as bilingual tasks where significant parallel corpora are available from WMT from previous years and the rest as monolingual, using solely English side of the corpora for deriving MTPP features. In accord, we treat en-de and ru-en as parallel MTPP and the rest as monolingual MTPP. RTM benefits from relevant data selection to be used as interpretants in both monolingual and bilingual settings. The related monolingual or bilingual datasets are used during feature extraction for the machine learning models of MT.

The machine learning models we use include ridge regression (RR), kernel ridge regression, support vector regression (SVR) (Boser et al., 1992), gradient tree boosting, extremely randomized trees (Geurts et al., 2006), and multi-layer perceptron (Bishop, 2006) as learning models in combination with feature selection (FS) (Guyon et al., 2002) and partial least squares (PLS) (Wold et al., 1984) where most of these models can be found in `scikit-learn`.[2] We experiment with:

- including the statistics of the binary tags obtained as features extracted from word-level tag predictions for sentence-level prediction,
- using RR to estimate the noise level for SVR, which obtains accuracy with 5% error compared with estimates obtained with known noise level (Cherkassky and Ma, 2004) and set $\epsilon = \sigma/2$.

We use Pearson's correlation (r), mean absolute error (MAE), root mean squared error (RMSE), relative absolute error (RAE), relative MAE (MAER), and mean RAE relative (MRAER) as evaluation metrics (Biçici and Way, 2015). Our best non-mix results are in Table 2 achieving 6th rank at best among 15 models in general.

3 Mixture of Experts Models

We use prediction averaging (Biçici, 2018) to obtain a combined prediction from various prediction outputs better than the components, where the performance on the training set is used to obtain

[2] `http://scikit-learn.org/`

		r_P	MAE	RMSE
Task 1	en-de	0.2622 (11)	0.5156 (8)	0.6828 (10)
	ru-en	0.6877 (8)	0.5138 (6)	0.6878 (7)
	en-zh	0.2310 (13)	0.5616 (6)	0.7298 (6)
	et-en	0.6067 (11)	0.5995 (8)	0.7284 (8)
	ne-en	0.5436 (11)	0.5308 (9)	0.6828 (9)
	si-en	0.5318 (10)	0.5003 (7)	0.6181 (7)
	ro-en	0.6990 (11)	0.5237 (8)	0.6574 (8)
Task 2	en-de	0.2289 (15)	0.1669 (15)	0.2081 (15)
	en-zh	0.3864 (15)	0.1585 (14)	0.1959 (15)

Table 2: RTM test results in sentence-level MTPP in tasks 1 and 2 using the best non-mix result with (ranks). r_P is Pearson's correlation.

weighted average of the top k predictions, $\hat{y}$ with evaluation metrics indexed by $j \in J$ and weights with w:

$$\begin{aligned} w_{j,i} &= \frac{w_{j,i}}{1-w_{j,i}} \\ \hat{\boldsymbol{y}}_{\mu_k} &= \frac{1}{k}\sum_{i=1}^{k}\hat{\boldsymbol{y}}_i && \text{MEAN} \\ \hat{\boldsymbol{y}}_{j,w_k^j} &= \frac{1}{\sum_{i=1}^{k} w_{j,i}}\sum_{i=1}^{k} w_{j,i}\,\hat{\boldsymbol{y}}_i \\ \hat{\boldsymbol{y}}_k &= \frac{1}{|J|}\sum_{j\in J}\hat{\boldsymbol{y}}_{j,w_k^j} && \text{MIX} \end{aligned} \quad (1)$$

We assume independent predictions and use $p_i/(1-p_i)$ for weights where p_i represents the accuracy of the independent classifier i in a weighted majority ensemble (Kuncheva and Rodríguez, 2014). We use the MIX prediction only when we obtain better results on the training set. We select the best model using r and mix the results using r, RAE, MRAER, and MAER. We filter out those results with higher than 0.875 relative evaluation metric scores.

We also use generalized ensemble method (GEM) as an alternative to MIX to combine using weights and correlation of the errors, $C_{i,j}$, where GEM achieves smaller error than the best combined model (Perrone and Cooper, 1992):

$$\begin{aligned} \hat{\mathbf{y}}_{\text{GEM}} &= \sum_{i=1}^{L} w_i\psi_i(\mathbf{x}) = \mathbf{y} + \sum_{i=1}^{L} w_i\epsilon_i \\ C_{i,j} &= E[\epsilon_i, \epsilon_j] = (\psi_i(\mathbf{x}) - \mathbf{y})^T(\psi_i(\mathbf{x}) - \mathbf{y}) \\ w_i &= \frac{\sum_{j=1}^{L} C_{i,j}}{\sum_{k=1}^{L}\sum_{j=1}^{L} C_{k,j}} \end{aligned}$$

Model combination (Figure 2) selects top k combined predictions and adds them to the set of predictions where the next layer can use another model combination step or just pick the best model according to the results on the training set. We use a two layer combination where the second layer is a combination of all of the predictions obtained. The last layer is an arg max.

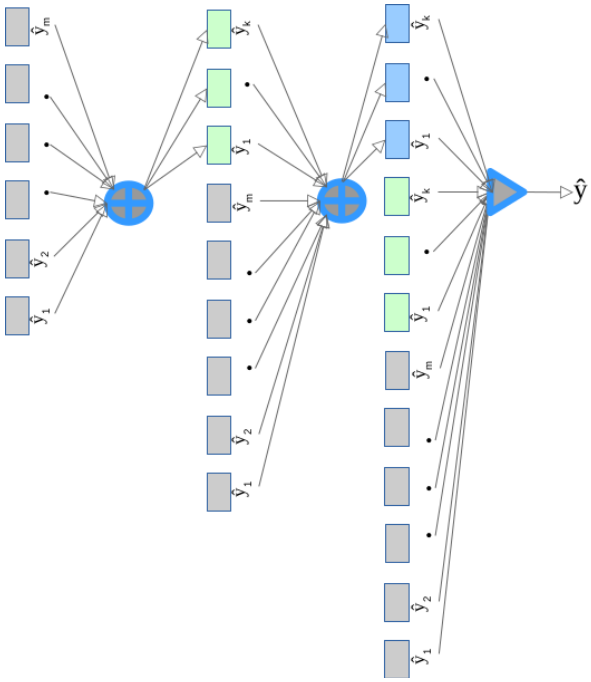


Figure 2: Model combination.

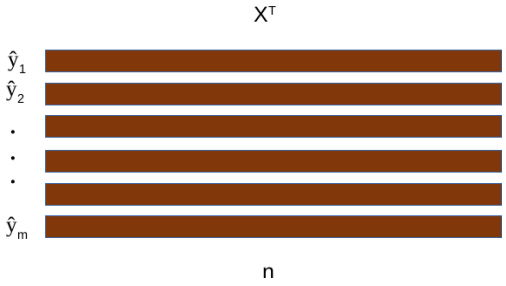


Figure 3: Stacking use predictions as features.

We also use stacking (STACK) to build higher level models using predictions from base prediction models where they can also use the probability associated with the predictions (Ting and Witten, 1999). The stacking models use the predictions from predictors as features and additional selected features and build second level predictors. Stacking with m predictors is depicted in Figure 3 where predictions are used as features for the predictors in the next level. Martins et al. (2017) used a hybrid stacking model to combine the word-level predictions from 15 predictors using neural networks with different initializations together with the previous features from a linear model. Our stacking results also use top features from the data similar to the pass through feature of the stacking regressor of sklearn.[3] For these features, we con-

[3] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.StackingRegressor.html

	r_P	trans	GEM mix	STACK
Task 1		en-de	0.2205	0.4244
		en-zh	0.43	0.5426
		et-en	0.5518	0.6245
		ne-en	0.537	0.6182
		si-en	0.4984	0.5907
		ro-en	0.7025	0.7518
		ru-en	0.7245	0.7734
Task 2		en-de	0.4023	0.5153
		en-zh	0.4124	0.5193

Table 3: RTM train results in sentence-level MTPP in tasks 1 and 2. r_P is Pearson's correlation.

		r_P	MAE	RMSE
Task 1	en-de	0.2804 (10)	0.5139 (8)	0.6762 (7)
	ru-en	0.7009 (7)	0.4957 (5)	0.6776 (5)
	en-zh	0.2310 (13)	0.5616 (6)	0.7298 (6)
	et-en	0.6051 (11)	0.5998 (8)	0.7268 (8)
	ne-en	0.6186 (9)	0.4990 (9)	0.6422 (8)
	si-en	0.5493 (10)	0.4909 (6)	0.6055 (6)
	ro-en	0.7367 (10)	0.4967 (7)	0.6167 (7)
	multi	0.5063 (8)	0.5249 (5)	0.6628 (6)
Task 2	en-de	0.2631 (15)	0.1601 (14)	0.1983 (15)
	en-zh	0.4029 (15)	0.1574 (14)	0.1933 (15)

Table 4: RTM test results in sentence-level MTPP in tasks 1 and 2 using the best GEM mix + mix result.

sider at most the top 15% of the features selected with feature selection.

RTM can achieve better results than the baseline model in Task 1 in all tasks participated [4] where the baseline is a neural predictor-estimator approach implemented in OpenKiwi (Kepler et al.). Our training r_P results are in Table 3. Our test set results using GEM mix and MIX are in Table 4 where we obtain 5th rank among 11 submissions in the multilingual subtask according to MAE. Official evaluation metric is r_P.

Before model combination, we further filter prediction results from different machine learning models based on the results on the training set to decrease the number of models combined and improve the results. A criteria that we use is MREAR ≥ 0.875 since MRAER computes the mean relative RAE score, which we want to be less than 1. In general, the combined model is better than the best model in the set and stacking achieves better results than MIX on the training set. However, stacking models significantly improve the results on the training data but obtain decreased scores on the test set (Table 5).

[4] Task1: https://competitions.codalab.org/competitions/24447#results, Task2: https://competitions.codalab.org/competitions/24515#results

		r_P	MAE	RMSE
Task 1	en-de	0.2289 (15)	0.6319 (13)	0.7754 (13)
	ru-en	0.6057 (8)	0.7526 (10)	0.9917 (10)
	en-zh	0.1504 (15)	0.8043 (11)	1.0249 (11)
	et-en	0.4014 (13)	1.1209 (13)	1.3892 (13)
	ne-en	0.4856 (13)	0.5662 (10)	0.7688 (10)
	si-en	0.3720 (14)	1.1118 (14)	1.2967 (14)
	ro-en	0.5858 (15)	1.4448 (15)	1.7387 (15)
Task 2	en-de	0.2387 (18)	0.2305 (17)	0.2896 (18)
	en-zh	0.2701 (20)	0.5008 (19)	0.5391 (20)

Table 5: RTM test results in sentence-level MTPP in tasks 1 and 2 using stacking.

4 Conclusion

Referential translation machines pioneer a language independent approach and remove the need to access any task or domain specific information or resource and can achieve top performance in automatic, accurate, and language independent prediction of translation scores. We present RTM results with ensemble models and stacking.

Acknowledgments

The research reported here received financial support from the Scientific and Technological Research Council of Turkey (TÜBİTAK) and Boğaziçi University, Turkey.

References

Ergun Biçici. 2018. RTM results for predicting translation performance. In *Proc. of the Third Conf. on Machine Translation (WMT18)*, pages 765–769, Brussels, Belgium.

Ergun Biçici. 2019. Machine translation with parfda, moses, kenlm, nplm, and pro. In *Proc. of the Fourth Conf. on Machine Translation (WMT19)*, Florence, Italy.

Ergun Biçici, Declan Groves, and Josef van Genabith. 2013. Predicting sentence translation quality using extrinsic and language independent features. *Machine Translation*, 27(3-4):171–192.

Ergun Biçici and Andy Way. 2015. Referential translation machines for predicting semantic similarity. *Language Resources and Evaluation*, pages 1–27.

Christopher M. Bishop. 2006. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg.

Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. 1992. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92, page 144–152, New York, NY, USA. Association for Computing Machinery.

Vladimir Cherkassky and Yunqian Ma. 2004. Practical selection of svm parameters and noise estimation for svm regression. *Neural Networks*, 17(1):113–126.

Pierre Geurts, Damien Ernst, and Louis Wehenkel. 2006. Extremely randomized trees. *Machine Learning*, 63(1):3–42.

Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. 2002. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1-3):389–422.

Kenneth Heafield, Ivan Pouzyrevsky, Jonathan H. Clark, and Philipp Koehn. 2013. Scalable modified Kneser-Ney language model estimation. In *51st Annual Meeting of the Assoc. for Comp. Ling.*, pages 690–696, Sofia, Bulgaria.

Fabio Kepler, Jonay Trénous, Marcos Treviso, Miguel Vera, and André F. T." Martins. OpenKiwi: An open source framework for quality estimation. In *Proc. of the 57th Annual Meeting of the Assoc. for Computational Linguistics: System Demonstrations", month = 7, year = 2019, address = Florence, Italy, publisher = Assoc. for Computational Linguistics, pages = 117–122,*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *45th Annual Meeting of the Assoc. for Comp. Ling.*, pages 177–180.

Ludmila I. Kuncheva and Juan J. Rodríguez. 2014. A weighted voting framework for classifiers ensembles. *Knowledge and Information Systems*, 38(2):259–275.

André F.T. Martins, Marcin Junczys-Dowmunt, Fabio N. Kepler, Ramón Astudillo, Chris Hokamp, and Roman Grundkiewicz. 2017. Pushing the limits of translation quality estimation. *Transactions of the Association for Comp. Ling.*, 5:205–218.

Michael Perrone and Leon Cooper. 1992. When networks disagree: Ensemble methods for hybrid neural networks. Technical report, Brown Univ. Providence RI Inst. for Brain and Neural Systems.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A Study of Translation Edit Rate with Targeted Human Annotation. In *Assoc. for Machine Translation in the Americas*.

Lucia Specia, Frédéric Blain, Marina Fomicheva, Erick Fonseca, Vishrav Chaudhary, Francisco Guzmán, and André FT Martins. 2020. Findings of the wmt 2020 shared task on quality estimation. In *Proc. of the Fifth Conf. on Machine Translation: Shared Task Papers*, Online.

Kai Ming Ting and Ian H. Witten. 1999. Issues in stacked generalization. *Journal of Artificial Intelligence Research*, 10:271–289.

S. Wold, A. Ruhe, H. Wold, and III Dunn, W. J. 1984. The collinearity problem in linear regression. the partial least squares (pls) approach to generalized inverses. *SIAM Journal on Scientific and Statistical Computing*, 5:735–743.

NJU's submission to the WMT20 QE Shared Task

Qu Cui and **Xiang Geng** and **Shujian Huang*** and **Jiajun Chen**
National Key Laboratory for Novel Software Technology, Nanjing University
{cuiq,gx}@smail.nju.edu.cn, {huangsj,chenjj}@nju.edu.cn

Abstract

This paper describes our system of the sentence-level and word-level Quality Estimation Shared Task of WMT20. Our system is based on the QE Brain, and we simply enhance it by injecting noise at the target side. And to obtain the deep bi-directional information, we use a masked language model at the target side instead of two single directional decoders. Meanwhile, we try to use the extra QE data from the WMT17 and WMT19 to improve our system's performance. Finally, we ensemble the features or the results from different models to get our best results. Our system finished fifth in the end at sentence-level on both EN-ZH and EN-DE language pairs.

1 Introduction

Quality Estimation (QE) is a task to predict the quality of translations without relying on any references. QE plays a critical role in machine translation to reduce human efforts, such as deciding whether a translation is good enough for post-editing and indicating what edits are needed. This paper describes our system of the Shared Task on Word and Sentence-Level (QE Tasks 2) at WMT20. With the post-edited translations, all the quality scores can be computed automatically by TER-COM (Snover et al., 2006).

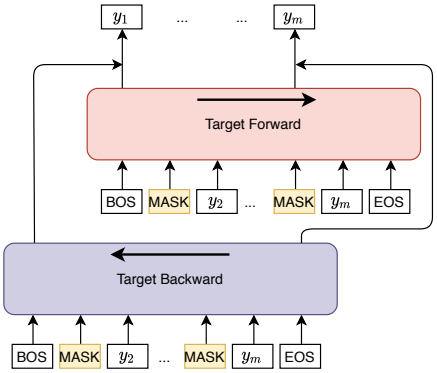
Traditional QE models (Kozlova et al., 2016) use some time-consuming and expensive hand-craft features to represent the translation pairs. With the great success of deep neural networks in natural language processing (NLP), some researches have begun to apply automatic neural features to do QE tasks (Chen et al., 2017; Shah et al., 2016). However, the rare QE data can't fully release the power of deep neural networks. To address this problem, researchers try to transfer bilingual knowledge from parallel data to QE tasks (Fan et al., 2018). These works usually follow a predictor-estimator framework (Kim et al., 2017). This framework first trains the predictor to predict each token of the target sentence given the source and the context of the target sentence on parallel data. Then, the estimator is trained using the features of QE data produced by the predictor.

However, existing predictor-estimator frameworks cannot fully use the information from parallel data because of the discrepancy of data quality between the predictor and the estimator. The predictor is trained on parallel data, which are nearly no errors in translations. While the translations in QE data is generated by a real machine translation system and may have some errors. When the estimator is training on the QE data, the predictor needs to extract the features of translations with some errors, which is quite different from the parallel data. Thus, the predictor can't extract features well.

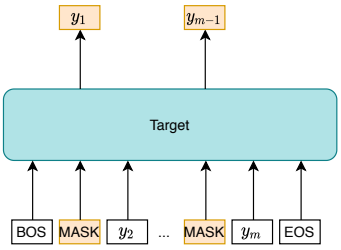
To fix this problem, we present two different approaches in this paper. The first model masks some tokens at the target side but still need to predict every token correctly, and it enhances the ability of the model to deal with translations with errors. And to obtain the deep bi-directional information, we use a masked language model at the target side instead of two single directional decoders. Meanwhile, we try to use the extra QE data, which are from the WMT17 and WMT19 to improve our system's performance. Finally, we ensemble the features or the results from different models to get our best results. Our system finished fifth in the end at sentence-level on both EN-ZH and EN-DE language pairs of the WMT20 QE shared tasks (Specia et al., 2020).

* Corresponding Author.

Proceedings of the 5th Conference on Machine Translation (WMT), pages 1004–1009
Online, November 19–20, 2020.



(a) QE Brain mask.



(b) MTLM.

Figure 1: To save space, we do not show the source encoders of these models in the figure. (a) shows the QE Brain mask system, and it simply enhances the original QE Brain system by simply masking tokens at the target side. (b) uses a masked language model at the target side to obtain deep bidirectional information.

2 Methods

As we all know, using different sub-models for ensemble will have better results (Krogh and Vedelsby, 1995). We ensemble different methods in our system, some of them are existing methods, and the others are proposed by us. Next, we will describe these methods.

2.1 Existing Methods

2.1.1 QUETCH

QUETCH (Kreutzer et al., 2015) (QUality Estimation from scraTCH) is a multilayer perceptron model trained without auxiliary parallel data. The embeddings of input passed through one linear layer with tanh activation functions and then one output layer with softmax activation functions, one linear layer with tanh activation functions, one output layer with softmax activation functions. QUETCH only outputs OK/BAD probabilities for each word in the word-level task. Similar to (Martins et al., 2017), we estimate HTER with the fraction of BAD labels for the sentence-level task.

2.1.2 NuQE

NuQE (Martins et al., 2016) (NeUral Quality Estimation) can be seen as a stronger version of QUETCH by using complex neural networks. The architecture of NuQE consists of one embeddings layer, one linear layer, one bi-directional GRU layer, two other linear layers. The input and output of NuQE is the same as QUETCH. We use QUETCH and NuQE as implemented in OpenKiwi (Kepler et al., 2019) [1].

2.1.3 QE Brain

QE Brain (Fan et al., 2018) is based on the predictor-estimator framework. The predictor uses transformer neural networks and will be pretrained on the parallel corpus. The model consists of encoder and bi-directional decoder to encode the source sentence $\mathbf{X} = \{x_1, x_2, \ldots, x_n\}$ and predict each token in the target sentence $\mathbf{Y} = \{y_1, y_2, \ldots, y_m\}$ with the help of hidden representations of the source sentence, respectively.

When training the Bi-LSTM (Graves and Schmidhuber, 2005), which is used as the estimator, the source sentence and translation are fed into the predictor to extract features. Similar to common predictor-estimator methods, QE Brain uses the hidden state of the final layer in the predictor as model derived features. They also extract the difference between the probability of generating the current token and the most likely token as mismatching features. Finally, the estimator concatenate model derived features and mismatching features to predict the word-level tags $\mathbf{O}$ and sentence-level HTER q.

Our proposed models are based on the QE Brain.

2.2 Proposed Methods

2.2.1 Masked QE Brain

Researches used to transfer bilingual knowledge from parallel data to QE tasks, however, the data distribution between parallel data and QE data is different. The translations in QE data are generated by a real machine translation system, and there will be some errors in these translations. While the translations in parallel data generated by humans, and there are nearly no errors. It means, the predictor trained on parallel data can not perform well when it is feeding with translations with errors because the contexts at the target side are different.

[1] https://unbabel.github.io/OpenKiwi.

Pair	Dataset	Train	Dev	Test
EN-DE	WMT20	7,000	1,000	1,000
	WMT19	13,442	-	-
	WMT17	23,000	-	-
EN-ZH	WMT20	7,000	1,000	1,000

Table 1: The statistics of QE dataset used in our system for the WMT20 QE shared task.

Pair	Train	Dev
EN-DE	23,438,059	2,000
EN-ZH	7,460,939	2,000

Table 2: Parallel Dataset statistics used in our system. We divide parallel data into a training set and development set.

To partially alleviate this problem, we proposed the masked QE Brain, as shown in Figure 1(a).

The motivation for our method is simple. We want to enhance the predicting ability of the model in the wrong contexts. To achieve this goal, when training the predictor on parallel data, we mask some tokens in the translation. And the predictor needs to make the same prediction as they are feeding with the complete pair. The other part is the same as the original version of the QE Brain.

2.2.2 Masked Target Language Model

The QE Brain and Masked QE Brain use a bi-directional decoder at the target side to obtain the information from both sides. However, this architecture is just a shallow concatenation which can not truly get the information from both sides (Devlin et al., 2018).

Thanks to the masked tokens in target sentences of Masked QE Brain, we can easily use a masked language model (Devlin et al., 2018) at the target side instead. We call this model the Masked Target Language Model (MTLM), and the format of the input is just the same as Masked QE Brain, as shown in Figure 1(b). They both input the source sentence $\mathbf{X}$, the masked target sentence $\mathbf{Y}'$. And the MTLM only need to predict the right tokens of these masked ones at the target side while Masked QE Brain needs to predict all the tokens.

3 Experiments

3.1 Dataset

3.1.1 Data statistics

QE Dataset The QE tasks of WMT20 contains both EN-DE language pair and EN-ZH language pair. They both have sentence-level and word-level tasks. Meanwhile, the word-level task contains the prediction for source tokens, target tokens, and target taps. In our paper, we only report word-level results on target tokens. In our work, we also use the EN-DE QE dataset of WMT17 and WMT19 to help train an ensemble model. The statistics of QE datasets are shown in Table 1.

Parallel Dataset For the EN-DE language pair, we use the data officially released by the organizers. And for the EN-ZH language pair, we use the parallel data from the WMT18 EN-ZH translation task. The statistics of parallel datasets are shown in Table 2.

3.1.2 Preprocess

EN-DE We use BPE (Sennrich et al., 2015) to segment both the English and German texts, and the BPE step is set to 30,000. We learn the BPE code jointly but build the two vocabularies separately. The size of EN is 14,112; the size of DE is 23,458.

EN-ZH We also use BPE to segment English texts here, and the setting is the same as those in EN-DE. The final size is 34,466. For Chinese texts, we keep all the sentences in the original QE dataset, and then use jieba[2] to segment other Chinese sentences in the parallel dataset. We choose the top 40,000 tokens of the frequency as the vocabulary.

3.2 Settings

Metrics The metric of sentence-level QE is Pearson's Correlation Coefficient. And the metrics of word-level QE are F1-MULT (the products of both positive and negative examples) and Matthews's Correlation Coefficient.

Hyper-parameters

- NuQE. The hidden size is [400, 200, 100, 50].
- QUETCH. The hidden size is [100, 50].
- QE Brain. The predictor contains one encoder and two decoders of 6 layers with 512 hidden units. The estimator is a Bi-LSTM, and its hidden size is 512.

[2] https://github.com/fxsjy/jieba

Pair	Method	Sent-level Dev	Word-level Dev	
			F1-MULT	MCC
EN-DE	NuQE	30.75	37.63	27.41
	QUETCH	31.27	37.19	27.78
	QE Brain	48.70	34.68	28.74
	QE Brain mask	**53.34**	35.15	30.17
	MTLM	49.77	**39.68**	**33.99**
	f-ensemble	59.91	-	-
	r-ensemble	59.76	-	-
	v-ensemble	-	47.58	42.36
EN-ZH	NuQE	42.49	43.50	33.02
	QUETCH	42.97	31.60	30.83
	QE Brain	58.05	44.07	32.85
	QE Brain mask	58.97	46.55	36.50
	MTLM	**60.89**	**51.31**	**43.33**
	f-ensemble	66.02	-	-
	r-ensemble	62.13	-	-
	v-ensemble	-	51.81	45.33

Table 3: Results of WMT20. f-ensemble means we ensemble different methods by features, r-ensemble means we ensemble different methods by their results and v-ensemble means different methods vote for an ensemble result.

- QE Brain mask. It is all the same as the QE Brain.
- MTLM. The predictor contains one encoder and one decoder of 6 layers with 512 hidden units. And the estimator is the same as the QE Brain.

3.3 Singe Model Results

Table 3 shows the single model results of our system. Different models are using the same parallel data and only using the QE dataset of WMT20.

The NuQE and QUETCH are only trained on the QE dataset, while the other methods are also trained on extra parallel datasets. We can see that the performance of NuQE and QUETCH is far from that of these models that have extra bilingual knowledge.

Compare with the original QE Brain, our two proposed models can have a big improvement.

3.4 Data Ensemble

We train to enhance our system by using other QE datasets, mainly from WMT17 and WMT19. We only try this on the EN-DE language pair. As we can see in Table 4, if we use more QE data, the performance can get a big improvement easily.

3.5 Model Ensemble

We also try to ensemble different methods and finally get the best result. For sentence-level, we try two different ways. First, we use QE Brain, QE Brain mask, and MTLM as a feature extractor. The features from the three models will be combined and then used to predict the hter scores. Second, we simply collect the predictions of different methods on the training set, development set and test set. The training predictions will be feed into a dense layer and used to predict true hter score, development predictions will be used to early stop. Finally, we will use the trained dense layer to deal with the test predictions.

For word-level, we simply use voting to ensemble different models. The results are shown in Table 3.

3.6 Final Results

Table 5 shows our final results of WMT20 on the web pages. Our system does not contain predictions on target gaps on the word-level, so we just combine the results on gaps from NuQE and QUETCH and our results on target tokens to build the final result.

Method	Dataset	Sent-level Dev	Word-level Dev	
			F1-MULT	MCC
QE Brain	WMT20	48.70	34.68	28.74
	ensemble	53.44	39.04	35.04
QE Brain mask	WMT20	53.34	35.15	30.17
	ensemble	54.87	40.05	35.25
MTLM	WMT20	49.77	39.68	33.99
	ensemble	53.38	43.40	36.41

Table 4: Ensemble results of the WMT20 EN-DE language pair, we train the QE systems on the combination of WMT20, WMT19, and WMT17 dataset.

Pair	Sent-level	Word-level
EN-DE	61.81 (5th)	45.11 (6th)
EN-ZH	64.23 (5th)	55.13 (6th)

Table 5: Final results and rank of WMT20 on the web page, the sentence-level metric is Pearson's Correlation Coefficient, and the word-level metric is the Matthews's Correlation Coefficient.

4 Conclusion

This paper describes our system of the WMT20 QE shared task. Our work mainly follows the QE Brain. To bridge the gap between parallel data and QE data, we use a simple way to bring noise into target sentences of parallel data. And to achieve deep bidirectional information, we use a masked language model at the target side. Experiments show that our two-step approaches achieve improvements. Meanwhile, we try to train our models on more QE data with the same language pair and ensemble different methods through different ways to get our final results.

References

Zhiming Chen, Yiming Tan, Chenlin Zhang, Qingyu Xiang, and Mingwen Wang. 2017. Improving machine translation quality estimation with neural network features. In *Proceedings of the Second Conference on Machine Translation*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*.

Kai Fan, Jiayi Wang, Bo Li, Fengming Zhou, Boxing Chen, and Luo Si. 2018. "bilingual expert" can find translation errors. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Netw*, 18(5-6).

Fábio Kepler, Jonay Trénous, Marcos Treviso, Miguel Vera, and André F. T. Martins. 2019. Openkiwi: An open source framework for quality estimation.

Hyun Kim, Jong-Hyeok Lee, and Seung-Hoon Na. 2017. Predictor-estimator using multilevel task learning with stack propagation for neural quality estimation. In *Proceedings of the Second Conference on Machine Translation*.

Anna Kozlova, Mariya Shmatova, and Anton Frolov. 2016. Ysda participation in the wmt'16 quality estimation shared task. In *Proceedings of the First Conference on Machine Translation*.

Julia Kreutzer, Shigehiko Schamoni, and Stefan Riezler. 2015. QUality estimation from ScraTCH (QUETCH): Deep learning for word-level translation quality estimation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*.

Anders Krogh and Jesper Vedelsby. 1995. Neural network ensembles, cross validation, and active learning. In *Advances in neural information processing systems*.

André F. T. Martins, Ramón Astudillo, Chris Hokamp, and Fabio Kepler. 2016. Unbabel's participation in the WMT16 word-level translation quality estimation shared task. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*.

André Martins, Marcin Junczys-Dowmunt, Fabio Kepler, Ramon Astudillo, Chris Hokamp, and Roman Grundkiewicz. 2017. Pushing the limits of translation quality estimation. *Transactions of the Association for Computational Linguistics*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual*

Meeting of the Association for Computational Linguistics (Volume 1: Long Papers).

Kashif Shah, Fethi Bougares, Loïc Barrault, and Lucia Specia. 2016. Shef-lium-nn: Sentence level quality estimation with neural network features. In *Proceedings of the First Conference on Machine Translation*.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. *Proceedings of Association for Machine Translation in the Americas*.

Lucia Specia, Frédéric Blain, Marina Fomicheva, Erick Fonseca, Vishrav Chaudhary, Francisco Guzmán, and André FT Martins. 2020. Findings of the wmt 2020 shared task on quality estimation. In *Proceedings of the Fifth Conference on Machine Translation: Shared Task Papers*.

BERGAMOT-LATTE Submissions for the WMT20 Quality Estimation Shared Task

Marina Fomicheva,[1*] Shuo Sun,[2*] Lisa Yankovskaya,[3*] Frédéric Blain,[1]
Vishrav Chaudhary,[5] Mark Fishel,[3] Francisco Guzmán,[5] Lucia Specia[1,4]
[1]University of Sheffield, [2]Johns Hopkins University, [3]University of Tartu,
[4]Imperial College London, [5]Facebook AI
[1]{m.fomicheva,f.blain,l.specia}@sheffield.ac.uk
[2]ssun32@jhu.edu [3]{lisa.yankovskaya,fishel}@ut.ee
[5]{fguzman,vishrav}@fb.com

Abstract

This paper presents our submission to the WMT2020 Shared Task on Quality Estimation (QE)[1]. We participate in `Task 1` and `Task 2` focusing on sentence-level prediction. We explore (a) a black-box approach to QE based on pre-trained representations; and (b) glass-box approaches that leverage various indicators that can be extracted from the neural MT systems. In addition to training a feature-based regression model using glass-box quality indicators, we also test whether they can be used to predict MT quality directly with no supervision. We assess our systems in a multilingual setting and show that both types of approaches generalise well across languages. Our black-box QE models tied for the winning submission in four out of seven language pairs in `Task 1`, thus demonstrating very strong performance. The glass-box approaches also performed competitively, representing a lightweight alternative to the neural-based models.

1 Introduction

Quality Estimation (QE) (Blatz et al., 2004; Specia et al., 2009) is an important part of Machine Translation (MT) pipeline. It allows us to evaluate how good a translation is without comparison to reference sentences. As part of the WMT20 Shared Task on Quality Estimation, two sentence-level tasks were proposed. In `Task 1`, participants are asked to predict human judgements of MT quality generated following a methodology similar to Direct Assessment (DA) (Graham et al., 2017). The goal of `Task 2` is to estimate the post-editing effort required in order to correct the MT outputs and measured using the HTER metric (Snover et al., 2006).

This year's task is different from the previous years in two important aspects: (i) the data includes seven language pairs, which are very different both typologically and in terms of translation quality; and (ii) the participants were provided with neural MT (NMT) models that were used for translation. We take advantage of this set up to compare black-box and glass-box approaches to QE. Furthermore, we test both approaches in a multilingual setting.

The rest of this paper is organised as follows. Section 2 describes the glass-box (2.1) and black-box (2.2) QE methods that we explore in our submissions. Section 3 describes the dataset used for the WMT2020 Shared Task on Quality Estimation. Section 4 provides our experimental settings, whereas Section 5 presents the results. Conclusions are given in Section 6.

2 Approach

Below we first describe our glass-box submissions based on the quality indicators that can be obtained as a by-product of decoding with an NMT system. Second, we present our neural-based QE submissions, which explore transfer learning with pre-trained representations. In both cases, we describe how QE is addressed as a multilingual task.

2.1 Glass-box

Glass-box approaches to QE are based on information from the NMT system used to translate the sentences, rather than looking at source and target sentences as in black-box QE, or using external resources. We rely on our previous work on glass-box QE that explores NMT output distribution to capture predictive uncertainty as a proxy to MT quality. Specifically, we use three groups of unsupervised quality indicators from Fomicheva et al. (2020).

[1] http://www.statmt.org/wmt20/quality-estimation-task.html

[*] Equal contribution.

Proceedings of the 5th Conference on Machine Translation (WMT), pages 1010–1017

Probability Features These features are based on the output probability distribution from a deterministic NMT system:

- Average word-level log-probability for the translated sentence (**TP**);
- Variance of word-level log-probabilities (**Sent-Var**); and
- Entropy of the softmax output distribution (**Softmax-Ent**).

Dropout Features This group of features also rely on output probability distribution but use uncertainty quantification based on the Monte Carlo dropout method to get more accurate QE results. This method consists of performing several forward passes through the network with parameters perturbed by dropout, collecting posterior probabilities and using the resulting distribution to estimate predictive uncertainty (Gal and Ghahramani, 2016).

- Expectation (**D-TP**) and variance (**D-Var**) over the NMT log-probability generated with Monte Carlo dropout;
- A ratio of **D-TP** and **D-Var** as described in Fomicheva et al. (2020) (**D-Combo**); and
- Lexical similarity between MT hypotheses generated with Monte Carlo dropout (**D-Lex-Sim**).

Attention Features We compute the entropy of encoder-decoder attention weights for each target token and then average token-level entropies to obtain a sentence-level measure. Given that the NMT systems used to generate the translations are based on the Transformer architecture where attention is computed at multiple layers and attention heads, there are [Layers $\times$ Heads] of averaged entropies for each sentence. Fomicheva et al. (2020) summarise them by taking the average or minimum value to obtain an unsupervised attention-based metric. By contrast, here we use the averaged entropies of attention weights coming from each head and layer combination as features in our regression model.

Algorithms We use the above groups of features as input for Random Forest (Ho, 1995) and XGBoost (Chen and Guestrin, 2016) regression algorithms. We also submitted the two best performing indicators from Fomicheva et al. (2020) with no supervision: **D-TP** and **D-Lex-Sim**.

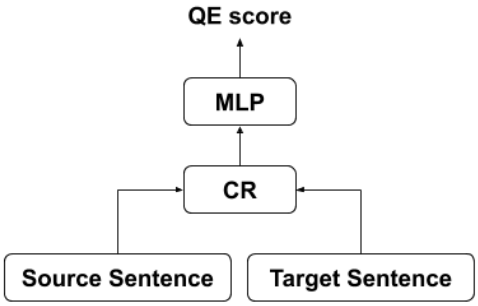


Figure 1: Black-box QE model built on top of contextualised representations (CR).

Multilinguality We hypothesise that system-internal indicators described above are by and large independent on the language pair, given that no linguistic information is directly used. Therefore, to build a multilingual QE system, i.e. a single model that can be used to predict quality for multiple language pairs, we simply concatenate the available data for all languages and use it for training our regression models. Note that we do not add any language identification markers and the system does not require them for making predictions. This can be useful for multilingual translation systems where the user does not need to identify the input languages, and especially for zero-shot settings where a given language pair may not have been seen at training time.

2.2 Black-box

We explore a baseline neural QE model and a multitask learning QE model, both of which are built on top of pre-trained contextualised representations (CR) such as BERT (Devlin et al., 2019) and XLM-R (Conneau et al., 2020).

Baseline QE model (BASE) Given a source sentence s^X in language X and a target sentence s^Y in language Y, we model the QE function f by stacking a 2-layer multilayer perceptron (MLP) on the vector representation of the [CLS] token from a contextualised representations model (CR):

$$\begin{aligned} f(s^X, s^Y) = & W_2 \cdot ReLU(\\ & W_1 \cdot E_{cls}(s^X, s^Y) + b_1 \\ &) + b_2 \end{aligned} \quad (1)$$

where $W_2 \in \mathbb{R}^{1\times 4h}$, $b_2 \in \mathbb{R}$, $W_1 \in \mathbb{R}^{4h\times h}$ and $b_1 \in \mathbb{R}^{4h}$. E_{cls} is a function that extracts the vector representation of the [CLS] token after encoding the concatenation of s^X and s^Y with CR and

ReLU is the Rectified Linear Unit activation function. Note that h is the output dimension of E_{cls}. We explore two training strategies: The **bilingual (BL)** strategy trains a QE model for every language pair while the **multilingual (ML)** strategy trains a single multilingual QE model for all language pairs, where the training data is simply pooled together without any language identifier. We note that this multilingual model here corresponds to a pooled, single-task learning approach.

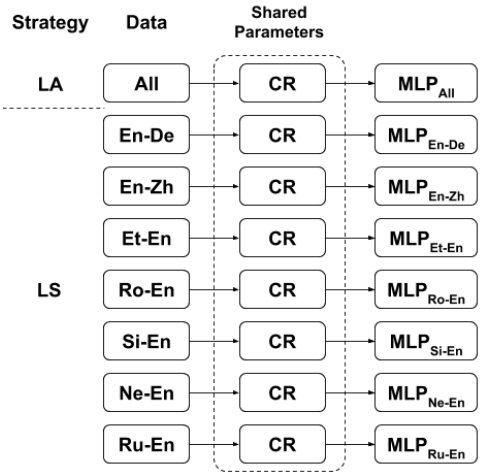


Figure 2: Multi-task learning QE model (MTL) with a shared BERT or XLM-R encoder.

Multi-task Learning QE Model (MTL) We explore multi-task learning to determine whether having parameter sharing across languages is beneficial, and to what degree having language-specific predictors can boost performance. We experiment with a multi-task approach where we concurrently optimise multiple QE BASE models that share parameters across languages. We jointly train two types of models: 1) language-specific (LS), which share parameters through a shared encoder but have different prediction layers; and 2) a language-agnostic (LA) model which also shares parameters for the prediction layer. We refer to these two models as **MTL-LA** and **MTL-LS**.

As seen in Figure 2, the MTL-LS submodels and MTL-LA submodel share a common BERT or XLM-R encoder, while each submodel has its own dedicated language-specific MLP. At training time, we iterate through the MTL-LS submodels in a round-robin fashion and alternate between training the MTL-LA submodel and training the chosen MTL-LS submodel. At test time, we can evaluate a test set with either the MTL-LA submodel or the MTL-LS submodel trained on the same language pair as the test set.

BiRNN We compared the above approaches to the BiRNN model from deepQuest (Ive et al., 2018). The BiRNN model uses an encoder-decoder architecture: it encodes both source and translation sentences independently using two bi-directional Recurrent Neural Networks (RNNs). The two resulting sentence representations are concatenated afterwards as the weighted sum of their word vectors, generated by an attention mechanism. For predictions at sentence-level, the weighted representation of the two input sentences is passed through a dense layer with sigmoid activation to generate the quality estimates. This is a light-weight variant of the black-box approaches above that does not rely on heavy pre-trained representations.

3 Data

This year two sentence-level QE tasks are available. For `Task 1` the participants are expected to predict DA-style human judgements (Graham et al., 2015), whereas the goal of `Task 2` is to estimate the post-editing effort (HTER). The data for `Task 1` includes six language pairs: Sinhala-English (Si-En), Nepalese-English (Ne-En), Estonian-English (Et-En), Romanian-English (Ro-En), English-German (En-De) and English-Chinese (En-Zh), where source sentences were extracted from Wikipedia articles. For `Task 2`, only English-Chinese and English-German are available. We also experimented with an additional dataset collected by IQT Labs in collaboration with the University of Sheffield. This is an Russian-English (Ru-En) dataset that contains a combination of Russian Reddit forums (75%) (using the Reddit API) and Russian WikiQuotes (25%). All MT outputs were generated by Transformer-based NMT systems (Vaswani et al., 2017). All datasets contain at least three DA judgements per MT segment by professional translators (0-100), with absolute quality scores standardised according to each annotator's mean and standard deviation. HTER labels were obtained by having professional translators fixing any errors in the translations, followed by using the TER[2] tool.

For each language pair the organisers provided training set (7000 sentences), development set (1000 sentences) and a blind test set (1000 sen-

[2] `http://www.cs.umd.edu/~snover/tercom/`

tences).

4 Settings

Glass-box To train proposed models, we used RandomForest from `sklearn` library[4] and XGBoost from `xgboost`[5] package. All input features are extracted from the NMT systems provided by the shared task's organisers. The number of features for `Probability` and `Dropout` groups does not depend on the parameters of the NMT systems and is equal to 3 and 4, respectively. The number of `Attention` features depends on the NMT system and is equal to the number of layers $\times$ the number of attention heads. We computed the sentence-level attention entropies in two ways: with and without the EOS token. For this reason, the total number of `Attention` features equals [Layers $\times$ Heads $\times$ 2]. This number is 96 for En-De/Zh and Et/Ro-En, and 192 for Si/Ne-En.

For our final experiments we combined the training (7000 sentences) and development (1000 sentences) sets, set a grid for the hyperparameters of our regression models and performed 5-fold cross-validation to choose the best hyperparameters.

Black-box We optimised our neural models with Adam (Kingma and Ba, 2015) and used the same learning rate ($1e^{-6}$) for all experiments. We trained each model on an Nvidia V100 GPU for 20 epochs with batch size of 8. Our final submission is an ensemble that combines the outputs from different variants of BASE and MTL QE models trained with different objective functions (mean squared error loss and huber loss) and contextualised encoder (BERT and XLM-R). We also included variants that use token-level log-probabilities from the NMT models as additional features. Each variant was trained 5 times with different random seeds. We used random forest (Breiman, 2001) to learn the ensemble. We set `n_estimators` to 500 and used the default values in `sklearn` for other hyperparameters.

[3]Results for the glass-box systems presented are slightly different from the official task results. The reason is that here we only show the results for the regression model trained with XGBoost, whereas both XGBoost and Random Forest models were submitted to the task.

[4]https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html

[5]https://xgboost.readthedocs.io/en/latest/python/index.html

5 Results

In this section, we present and analyse the results for our submissions to `Task 1`. We provide a general comparison of the glass-box and black-box systems and also look at some specific aspects of their performance.

5.1 Overall Results

Table 1 shows the results of our submissions to `Task 1`. Besides Pearson correlation for each language pair, column `Avg` shows the average correlation across language pairs for each presented model, which corresponds to "multilingual" subtask from the organisers.[6] Note that although it was not required for the multilingual task to have a single QE system serving multiple languages, we build such systems for our multilingual experiments. The last column in Table 1 shows the number of parameters for each model. In the case of glass-box systems this corresponds to the number of features.[7]

The first group of systems in Table 1 corresponds to the glass-box approach including the unsupervised metrics and feature-based regression models (see Section 2.1).[8] Feature-based systems include models trained on a single language pair (Mono-LP), models based on multiple language pairs (Multi-LP) and an ensemble based on models trained with different amounts of data (see discussion below). The next group of systems corresponds to the black-box approach presented in Section 2.2. Besides the models based on pre-trained representations, we include BiRNN, a light-weight neural-based QE model.

The last two rows in Table 1 show the results of the baseline models prepared by the organisers and the Top #1 model. The baseline system is a neural predictor-estimator model trained with the default parameters described in OpenKiwi (Kepler et al., 2019). The predictor model was trained on the parallel data used to train the NMT models.

[6]The multilingual sub-task did not include Ru-En and it was not considered for the `Avg` column.

[7]As explained in Section 4, the number of features for the glass-box regression models changes depending on the language, as the corresponding NMT systems have different number of layers and attention heads. Thus, we have 199 features for Si/Ne-En, and 103 features for the rest of the language pairs.

[8]These experiments do not include Russian-English, as the corresponding NMT system is an ensemble and it is not evident how the glass-box features proposed by Fomicheva et al. (2020) should be extracted in this case.

		Et-En	Ro-En	Si-En	Ne-En	En-De	En-Zh	Ru-En	Avg	# Params
					Unsupervised					
Glass-box	D-TP	0.64	0.69	0.46	0.56	0.26	0.32	–	0.49	–
	D-Lex-Sim	0.61	0.67	0.51	0.60	0.17	0.31	–	0.48	–
					Regression					
	Mono-LP	0.68	0.79	0.56	0.66	0.46	0.43	–	0.60	103/199
	Multi-LP	0.68	0.79	–	–	0.45	0.41	–	0.58	103/199
	Ensemble	0.68	0.80	0.56	0.66	0.48	0.43	–	0.60	103/199
Black-box	BiRNN	0.33	0.50	0.39	0.35	0.10	0.18	–	0.31	13.3M
					BERT					
	BASE-BL	0.67	0.83	0.50	0.68	0.39	0.44	0.65	0.59	180M
	BASE-ML	0.70	0.85	0.53	0.69	0.42	0.45	0.65	0.61	180M
	MTL-LA	0.69	0.85	0.51	0.68	0.47	0.44	0.66	0.61	197M
	MTL-LS	0.69	0.84	0.51	0.69	0.47	0.45	0.65	0.61	197M
					XLM-R					
	BASE-BL	0.78	0.89	0.64	0.78	0.44	0.48	0.76	0.67	564M
	BASE-ML	0.80	0.89	0.67	0.78	0.50	0.49	0.78	0.69	564M
	MTL-LA	0.80	0.89	0.68	0.80	0.50	0.48	0.78	0.69	594M
	MTL-LS	0.81	0.89	0.66	0.80	0.51	0.49	0.77	0.69	594M
	Ensemble (BL)	**0.82**	**0.91**	**0.68**	0.81	**0.54**	0.53	0.80	–	
	Ensemble (ML)	**0.83**	**0.91**	**0.68**	0.81	**0.56**	0.53	–	0.72	–
	Baseline	0.48	0.69	0.37	0.39	0.15	0.19	–	0.38	
	Top #1	0.82	0.91	0.69	0.82	0.55	0.54	0.81	0.72	

Table 1: Results for `Task 1`: Pearson correlation coefficients between human DA scores and predicted values for WMT2020 test sets.[3] Avg is the average Pearson correlation across language pairs. Baseline and Top #1 results are taken from http://www.statmt.org/wmt20/quality-estimation-task_results.html. Results that are not significantly different from the Top #1 submission are marked in bold. We submitted results from ensemble (ML) to the multilingual subtask and results from ensemble (BL) to the per-language subtasks.

Below we summarize our observations:

General performance First, we observe that all our submitted systems outperform the baseline. In particular, the ensemble of models based on pre-trained contextualised representations achieves a very strong performance for some language pairs. It is either the top system or perform on par with the Top #1 submission, with no significant difference for Et-En, Ro-En, Si-En and En-De.[9]

Black-box models We also note that our XLM-R based models achieve a higher correlation with human judgements than the models built on top of BERT pre-trained representations, which can be related to the fact that XLM-R is a more powerful model with a much higher number of parameters. BiRNN, a light-weight neural-based QE system that does not use language model pre-training, shows lower correlation values, probably due to a relatively small amount of data available for training.

Glass-box models We note that glass-box systems perform competitively compared to some of the neural-based approaches. Interestingly, even the unsupervised submissions that rely only on the information extracted from the NMT models outperform the BiRNN and Predictor-Estimator neural-based QE systems, thus highlighting the benefit of this approach in a setting where a light-weight model is required (thus disallowing the use of BERT-style models fine-tuned on the QE task) and the amount of available training data is small. Regression-based models always improve on the individual unsupervised features for all language pairs (see Section 5.4 for discussion) and achieve comparable results to the BERT-based black-box systems.

5.2 Does model ensembling improve performance?

Ensembling multiple models is known to boost performance. We test whether this method improves the results for our systems. To produce ensemble for the glass-box approach, we computed an average of the predictions from the models trained with different amounts of data (see Section 5.5). As shown in Table 1, there is no difference between ensemble and individual models. For the black-box approach ensemble is produced by combining various types of models as described in Section 4. The ensemble of neural models provides a significant

[9] Here and in what follows we use the Hotelling-Williams test (Williams, 1959) to compute significance of the difference between dependent correlations with p-value < 0.05.

boost in performance at the cost of a very large number of parameters.

5.3 Multilingual models

For some MT production scenarios it is more convenient to have one multilingual QE model instead of having one model per language pair. We test how well the QE systems discussed in this paper perform in a multilingual setting. For the glass-box approach, we concatenated all training and development sets for En-De/Zh and Et/Ro-En together and trained a single model using this data. We exclude Si/Ne-En as we have a different number of features for these language pairs (see Section 4). Multilingual systems for the black-box approach are described in Section 2.2.

As can be seen from Table 1, both the glass-box and the black-box multilingual systems obtained results comparable to the models trained for individual language pairs. Thus, for the purposes of QE task both glass-box features and multilingual pre-trained representations generalise well across languages.

5.4 How does each group of features affect performance?

To investigate how each group of features affects performance of the glass-box models, we trained the models separately with different groups of features and their combinations, and computed Pearson correlation coefficients between predicted scores and DA. For our experiments we have three groups of features `Dropout`, `Probability` and `Attention`, all combinations of two of them and the combination of all three groups. We also show the correlation for some of the individual features: (i) translation probability (TP) as one of the simplest things we can extract from an NMT system; and (ii) two best performing unsupervised QE indicators from Fomicheva et al. (2020): dropout translation probability (D-TP) and dropout lexical similarity (D-Lex-Sim) (see Section 2.1).

As can be seen from Table 2, the best results among the individual groups of features are obtained for either `Dropout` features (Et/Ro/Si/Ne-En and En-Zh) or `Attention` features (En-De/Zh). The combination of all three groups of features and the combination of `Dropout` and `Attention` showed the best results for all language pairs.

Table 2 also shows the benefit of using supervision: combining features with XGBoost generally leads to a better correlation than directly using the best-performing individual QE indicators without any training ('Unsup' rows).

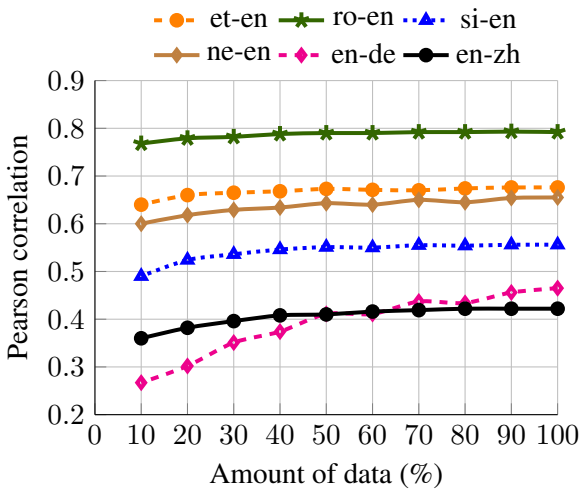


Figure 3: Pearson correlation coefficient between predicted values (glass-box models) of WMT2020 test sets and DA.

5.5 How many sentences do we need to train a QE system?

Here we investigate how the amount of available training data affects the performance of our systems. For this purpose, we randomly selected 10%, 20% ... 100% of the data and trained our models. We repeated data splitting and training of the models ten times; thus, we got 10 sets of predictions for each amount of data, we computed Pearson correlation coefficient between DA and predicted scores and took an average of these 10 correlation coefficients over each amount of data. As shown in Figure 3, the performance across the different amounts of training data with the glass-box models is stable for all language pairs except for En-De. Improvements over the best performing individual feature for each language pair can be obtained even with fairly small amounts of data.

Figure 4 shows the performance across different amounts of data for the BASE-BL black-box models. In this case, we observe larger improvements when more data is available for training. However, quite surprisingly, relatively high performance is achieved even with 5% and 10% of the data.

5.6 Task 2: HTER prediction

Besides experiments with DA labels, we used the same approach to train models with HTER data for En-De and En-Zh language pairs. Table 3 shows

		Et-En	Ro-En	Si-En	Ne-En	En-De	En-Zh
Type of features	Attention	0.519	0.722	0.455	0.583	0.382	0.353
	Dropout	**0.669**	0.751	0.548	0.638	0.206	0.352
	Probability	0.525	0.670	0.508	0.568	0.189	0.329
	Dropout+Probability	**0.670**	0.754	**0.556**	0.632	0.194	0.381
	Attention+Probability	0.611	0.700	**0.550**	0.629	**0.454**	0.406
	Attention+Dropout	**0.679**	**0.791**	**0.554**	**0.659**	**0.452**	**0.429**
	All	**0.678**	**0.793**	**0.556**	**0.657**	**0.464**	**0.427**
	Unsup:D-TP	0.642	0.693	0.460	0.558	0.259	0.321
	Unsup:D-Lex-Sim	0.612	0.669	0.513	0.600	0.172	0.313
	Unsup:TP	0.486	0.647	0.399	0.482	0.208	0.257

Table 2: Pearson correlation coefficients between human DA scores and predicted values for WMT2020 test sets. The unsupervised results are taken from (Fomicheva et al., 2020). Results marked in bold are not significantly outperformed by any other method.

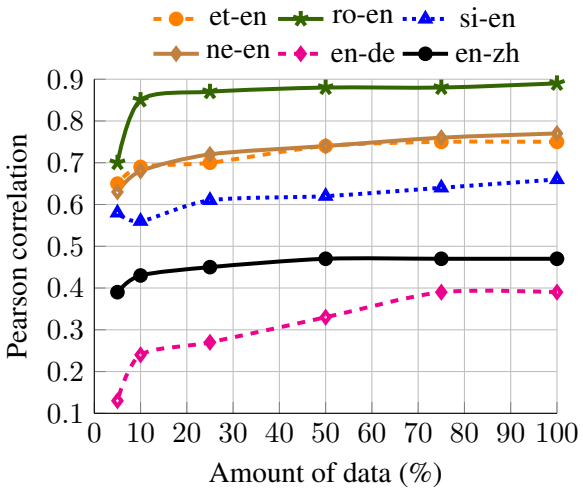


Figure 4: Pearson correlation coefficient between predicted values (black-box BASE-BL models) of WMT2020 dev sets and DA.

		En-De	En-Zh
Glass-box	Mono-LP	0.601	0.605
	Ensemble	0.613	0.613
	Baseline	0.392	0.506
	Top #1	0.758	0.664

Table 3: Results for `Task 2` (sentence-level): Pearson correlation coefficient between HTER and predicted values for WMT2020 test set. The results of Baseline and the best models are taken from `http://www.statmt.org/wmt20/quality-estimation-task_results.html`.

Pearson correlation between the predictions and HTER scores for glass-box systems.[10] Interestingly, the glass-box approach performs more competitively when predicting HTER than when estimating DA scores, as the gap between our submission and the best performing system is smaller. Thus, this type of judgements might be easier to predict based on system-internal information from NMT models.

6 Conclusions

We presented glass-box and black-box models submitted to the WMT2020 QE shared task. Black-box models showed the results on a par with the top submissions. Glass-box methods achieve from moderate to strong linear correlation with human judgments and can be used as a light-weight and cost-effective alternative in a scenario where the NMT model is available. Besides that, we conducted experiments to test the performance of our QE systems in a multilingual setting. We showed that the performance of both approaches is comparable when training and predicting on the same language pair, and when training a single model to predict on multiple language pairs.

Acknowledgements

Marina Fomicheva, Lisa Yankovskaya, Frédéric Blain, Mark Fishel, and Lucia Specia were supported by funding from the Bergamot project (EU H2020 Grant No. 825303).

[10] We did not prepare neural-based QE systems for this task due to time limitations.

References

John Blatz, Erin Fitzgerald, George Foster, Simona Gandrabur, Cyril Goutte, Alex Kulesza, Alberto Sanchis, and Nicola Ueffing. 2004. Confidence estimation for machine translation. In *Proceedings of the 20th international conference on Computational Linguistics*, page 315. Association for Computational Linguistics.

Leo Breiman. 2001. Random forests. *Machine learning*, 45(1):5–32.

Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Marina Fomicheva, Shuo Sun, Frédéric Blain Lisa Yankovskaya, Francisco Guzmán, Mark Fishel, Nikolaos Aletras, Vishrav Chaudhary, and Lucia Specia. 2020. Unsupervised quality estimation for neural machine translation.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In *International Conference on Machine Learning*, pages 1050–1059.

Yvette Graham, Timothy Baldwin, and Nitika Mathur. 2015. Accurate evaluation of segment-level machine translation metrics. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1183–1191.

Yvette Graham, Timothy Baldwin, Alistair Moffat, and Justin Zobel. 2017. Can machine translation systems be evaluated by the crowd alone. *Natural Language Engineering*, 23(1):3–30.

Tin Kam Ho. 1995. Random decision forests. In *Document analysis and recognition, 1995., proceedings of the third international conference on*, volume 1, pages 278–282. IEEE.

Julia Ive, Frédéric Blain, and Lucia Specia. 2018. Deepquest: a framework for neural-based quality estimation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3146–3157.

Fábio Kepler, Jonay Trénous, Marcos Treviso, Miguel Vera, and André FT Martins. 2019. Openkiwi: An open source framework for quality estimation. *arXiv preprint arXiv:1902.08646*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of association for machine translation in the Americas*, volume 200.

Lucia Specia, Marco Turchi, Nicola Cancedda, Marc Dymetman, and Nello Cristianini. 2009. Estimating the sentence-level quality of machine translation systems. In *13th Conference of the European Association for Machine Translation*, pages 28–37.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Evan James Williams. 1959. *Regression Analysis*, volume 14. Wiley, New York, USA.

The NiuTrans System for the WMT20 Quality Estimation Shared Task

Chi Hu[†] **Hui Liu**[†] **Kai Feng**[†] **Chen Xu**[†] **Zefan Zhou**[†] **Shiqin Yan**[†]
Yingfeng Luo[†] **Chenglong Wang**[†] **Xia Meng**[†] **Nuo Xu**[†]
Tong Xiao[†‡] **Jingbo Zhu**[†‡]
[†]NLP Lab, School of Computer Science and Engineering
Northeastern University, Shenyang, China
[‡]NiuTrans Reasearch, Shenyang, China
huchinlp@gmail.com, liuhui0717@outlook.com,
{xiaotong,zhujingbo}@mail.neu.edu.cn,

Abstract

This paper describes the submissions of the NiuTrans Team to the WMT 2020 Quality Estimation Shared Task (Specia et al., 2020). We participated in all tasks and all language pairs. We explored the combination of transfer learning, multi-task learning and model ensemble. Results on multiple tasks show that deep transformer machine translation models and multilingual pretraining methods significantly improve translation quality estimation performance. Our system achieved remarkable results in multiple level tasks, e.g., our submissions obtained the best results on all tracks in the sentence-level Direct Assessment task[1].

1 Introduction

Quality estimation (QE) evaluates the quality of machine translation output without human reference translations (Blatz et al., 2004). It has a wide range of applications in post-editing and quality control for machine translation.

We participated in all tasks and language pairs at the WMT 2020 QE shared task[2], including sentence-level Direct Assessment tasks, word and sentence-level post-editing effort tasks, and document-level QE tasks. We investigated transfer learning and ensemble methods using recently proposed multilingual pre-trained models (Devlin et al., 2019; Conneau et al., 2020) as well as deep transformer models (Wang et al., 2019a). Our main contributions are as follows:

- We apply multi-phase pretraining (Gururangan et al., 2020) methods under both high- and low-resource settings to QE tasks.
- We incorporate deep transformer NMT models into QE models.
- We propose a simple strategy to convert document-level tasks into word- and sentence-level tasks.
- We explore effective ensemble methods for both word- and sentence-level predictions.

Results on different level tasks show that our methods are very competitive. Our submissions achieved the best Pearson correlation on all language pairs of the sentence-level Direct Assessment task and the best results on English-Chinese post-editing effort tasks.

We present methods for the sentence-level Direct Assessment task in §2. Then in §3 and §4, we describe our approaches to post-editing tasks and document-level tasks, respectively. System ensemble methods are discussed in §5. We show the detail of our submissions and the results in §6. We conclude and discuss future work in §7.

2 Sentence-level Direct Assessment Task

The sentence-level Direct Assessment task is a new task where sentences are annotated with Direct Assessment (DA) scores by professional translators rather than post-editing labels. DA scores for each sentence are rated from 0 to 100, and participants are required to score sentences according to z-standardized DA scores. The DA task consists of seven tracks for different language pairs and one multilingual track. Submissions were evaluated in terms of Pearson's correlation metric for the DA prediction against human DA (z-standardized mean DA score, i.e., z-mean).

2.1 Datasets and Resources

This task contains 7K sentences for training and 1K sentences for development on each language pair,

[1]Our number of submissions exceeded the daily or total limit.

[2]http://www.statmt.org/wmt20/quality-estimation-task.html

Proceedings of the 5th Conference on Machine Translation (WMT), pages 1018–1023
Online, November 19–20, 2020.

including sentence scores and word probabilities from the NMT models. The organizer also provided parallel data used to train the NMT models except for Russian-English, ranging from high resource (En-De, En-Zh), medium resource (Ro-En), to low-resource (Et-En, Ne-En, Si-En).

In addition to the official data, we also used some multilingual pre-trained models for fine-tuning, including multilingual BERT[3] (mBERT) and XLM-RoBERTa[4] (XLM-R).

2.2 Unsupervised Quality Estimation

Our baseline system was built upon unsupervised quality estimation methods proposed by Fomicheva et al. (2020), which use out-of-box NMT models as sources of information for directly estimating translation quality. We utilized the output sentence probabilities from NMT models as indicators for QE tasks. Given the input sequence $\mathbf{x}$, suppose the decoder generates an output sequence $\mathbf{y} = y_1, \ldots, y_T$ of length T, the probability of generating $\mathbf{y}$ is factorized as:

$$p(\mathbf{y}|\mathbf{x}, \theta) = \prod_{t=1}^{T} p\left(y_t|\mathbf{y}_{<t}, \mathbf{x}, \theta\right) \quad (1)$$

where θ represents model parameters. The output probability distribution $p\left(y_t \mid \mathbf{y}_{<t}, \mathbf{x}, \theta\right)$ is produced by the decoder over the *softmax function*.

We considered the sequence-level translation probability normalized by length:

$$\text{TP} = \frac{1}{T} \sum_{t=1}^{T} \log p\left(y_t|\mathbf{y}_{<t}, \mathbf{x}, \theta\right) \quad (2)$$

And the probability generated from perturbed parameters with dropout, we performed N times inference and used the averaged output:

$$\text{D-TP} = \frac{1}{N} \sum_{n=1}^{N} \text{TP}_{\hat{\theta}^n} \quad (3)$$

2.3 Multi-phase Pretraining

Fine-tuning pre-trained language models have become the foundation of today's NLP (Devlin et al., 2019; Conneau et al., 2020). Recent advances in pre-trained multilingual language models lead to state-of-the-art results on QE tasks (Kim et al., 2019; Kepler et al., 2019a). Similar to Gururangan et al. (2020), we continued training multilingual pre-trained models in both domain- and task-adaptive manners.

Domain-adaptive pretraining uses a straightforward approach–we continue pretraining mBERT and XLM-R on the parallel corpora provided by the organizers, which is used to train the MT systems. Unlike the training data labeled with DA scores, the parallel data for different language pairs vary. The corpus of pre-trained language models also has the problem of data imbalance. In practice, we increased the training frequency of low-resource data.

Task-adaptive pretraining refers to pretraining on the unlabeled training set for a given task. Compared to domain-adaptive pretraining, it uses a far smaller corpus, but the data is much more task-relevant. We used the same models as the domain-adaptive pretraining.

2.4 Fine-tuning

Similar to previous work (Kepler et al., 2019a; Yankovskaya et al., 2019), we used models trained with the above methods as feature extractors for the sentence-level scoring tasks. We treated the scoring task as a regression task. Following standard practice, we added a separator token between source and target sentences and passed the pooled representation from the encoder to a task-specific feed-forward layer for classification. We used the z-standardized mean DA score as the ground truth and minimized the mean squared error during training.

3 Word and Sentence-Level Post-editing Effort Task

This task consists of the word- and sentence-level tracks to evaluate post-editing effort. The word-level tasks predicts OK or BAD tags in both source and target sequences. It evaluates the Matthews correlation coefficient[5] (MCC) for tags. The sentence-level task predicts HTER scores, which is the ratio between the number of edits needed and the reference translation length. It evaluates Pearson's correlation for the HTER prediction. There are two language pairs in both the word- and sentence-level tasks, including English-German (En-De) and English-Chinese (En-Zh).

[3] https://huggingface.co/bert-base-multilingual-cased

[4] https://github.com/facebookresearch/XLM

[5] https://en.wikipedia.org/wiki/Matthews_correlation_coefficient

3.1 Datasets and Resources

The labeled data consists of 7K sentences for training and 1K sentences for development for each language pair. We used the additional parallel data provided by the organizers to train predictors, containing about 20M En-Zh sentence pairs and 23M En-De sentence pairs after pre-processing with the NiuTrans SMT toolkit (Xiao et al., 2012). Pretrained language models include mBERT and XLM-R, were also used for Task 2.

3.2 Predictor-Estimator Models

The predictor-estimator architecture and its variants (Kim et al., 2017; Kepler et al., 2019b) had established state-of-the-art on WMT QE tasks. The system consists of a word prediction module (predictor) trained from additional large-scale parallel corpora and a quality estimation module (estimator) trained from quality-annotated data.

For the sentence-level tasks and target-side word-level tasks, we employed the official bi-RNN predictor-estimator trained with `OpenKiwi` (Kepler et al., 2019b) as the baseline. Similar to Wang et al. (2019b), we used NMT models trained with back-translation as predictors.

The original predictor and estimator use RNNs to encode the source and predict tags or scores. We also implemented two transformer-based predictors which replace the RNN with transformer (Vaswani et al., 2017) or deep transformer architectures (Wang et al., 2019a; Li et al., 2019). We compared different tokenizing strategies such as word segmentation and byte pair encoding (BPE) (Sennrich et al., 2016) for all language pairs.

3.3 Multi-task learning

The word- and sentence-level tasks are highly related to their annotations are commonly based on the HTER measure. We used a linear summation of sentence-level and target word-level objective losses as follows:

$$\mathcal{L} = \mathcal{L}_{mt.word} + \mathcal{L}_{mt.gap} + \mathcal{L}_{HTER} \quad (4)$$

where the components denote the loss of target-word, target-gap, and predictions for HTER score.

We also trained models using source sentence and origin/post-edited MT output to predict the source-side word level tags:

$$\mathcal{L}_{SRC} = \mathcal{L}_{src-mt} + \mathcal{L}_{src-pe} \quad (5)$$

4 Document-Level QE Task

This task aims to predict document-level quality scores as well as fine-grained annotations. Each document is annotated for translation errors with word span, severity, and error type[6]. Additionally, there are document-level scores (MQM scores) generated from the error annotations using the method proposed by Torrón and Koehn (2016). The annotation task evaluates F1 scores on the gold annotations. The scoring task evaluates the Pearson's correlation between the gold and predicted MQM scores.

4.1 Datasets and Resources

We also used 35M WMT14 En-Fr parallel data to train our predictors for the annotation task except for the official 1,448 En-Fr documents. For the scoring task, we used pre-trained language models, including mBERT and XLM-R.

4.2 Document-level Annotating Task

Following Kepler et al. (2019a), we treated the document-level annotation problem as a word-level task, with each sentence processed separately. We tokenized the training set and tagged each token with an OK/BAD tag. Specifically, each token was labeled as BAD if it contains any character in error spans. Besides token tags, we labeled a gap as BAD if a span begins and ends exactly in its borders. Otherwise, it was labeled as OK. During the test time, we mapped BAD tags to annotations in a single scheme: (a) continuous labels were merged into an error annotation; (b) individual labels were directly converted to error annotations. We ignored the severity information and always treated the error as the most frequent 'major'.

We adopt the predictor-estimator architecture for this task. We implemented our predictors with deep transformers with relative position representation. The settings for model training are described in (Hu et al., 2020). We also compared two tokenization schemes, including word-level tokenization and BPE. Similar to Task 2, we jointly trained our models with target-side word-level and word gap tasks.

4.3 Document-level Scoring Task

We treated the document-level scoring task as a sentence-level task with a simple mapping scheme.

[6] http://www.qt21.eu/mqm-definition/definition-2015-12-30.html

We also ignored all critical and minor errors, and thus the MQM score for each document is calculated by:

$$\text{MQM} = 100 \times (1 - \frac{W \times Count_{major}}{Count_{word}}) \quad (6)$$

where $Count_{major}$ and $Count_{word}$ are the count of major errors and total words, respectively. W denotes the weight of major errors, which is fixed at 5 in our experiments.

Then we score each sentence according to the number of errors it contains:

$$\text{Score}_{\text{sent}} = 100 - W \times Count_{major} \quad (7)$$

We applied the same fine-tuning strategies, as mentioned in Sec 2, to this task. During the test time, the count of errors was retrieved from the predicted score of all sentences. A document score is 0 if it has too many errors.

5 System Ensemble

In addition to training models for each task, we also explored effective ensemble methods to combine outputs for different level tasks.

5.1 Word-level ensemble

We used two approaches to ensemble word-level predictions for Task 2 and Task 3.

Voting-Based Ensemble. Voting is the easiest method to combine predictions from multiple models. We chose the label with the most votes for each token as the output.

Averaging-Based Ensemble. Similar to Kepler et al. (2019a), we used Powell's conjugate direction method to optimize the task metric (MCC or F1 score) and learn the weights of different systems on the development set.

5.2 Sentence-level ensemble

We averaged the predicted scores from multiple models associated with different weights. The weights were also learned on the development set using Powell's method. We removed outliers from the candidate pool to make the prediction more stable.

6 Experiments and Results

6.1 Task 1

Below we describe our systems for Task 1.

Unsupervised baseline. As described in §2, our baseline system leverages the output probabilities from NMT models to assess the sentence score. We performed 20 inference passes and set the dropout rate as 0.3 for all language pairs.

Pair	TP Score	D-TP Score
En-De	0.249	0.273 (+10%)
En-Zh	0.330	0.348 (+5%)
Ro-En	0.648	0.693 (+7%)
Et-En	0.497	0.562 (+13%)
Ne-En	0.431	0.490 (+14%)
Si-En	0.423	0.469 (+11%)
Ru-En	0.518	0.535 (+3%)

Table 1: Pearson (r) correlation between unsupervised methods and human DA judgements on the validation data for sentence-level DA tasks. We mark improvements of D-TP by percentage.

Pair	mBERT	XLM-R	Ensemble
En-De	0.516	0.555	0.562
En-Zh	0.512	0.533	0.551
Ro-En	0.888	0.911	0.917
Et-En	0.809	0.820	0.833
Ne-En	0.816	0.821	0.830
Si-En	0.607	0.670	0.698
Ru-En	0.728	0.796	0.816
Multilingual	-	-	0.732

Table 2: Pearson (r) correlation between pretraining methods and human DA judgements on the test data for sentence-level DA tasks. We only present the results of XLM-R-large for the second method.

Pretraining and fine-tuning. We experimented with different pre-trained models for multi-phase pretraining and fine-tuning. Specifically, we used three model settings, including mBERT-base cased (∼200M parameters), XLM-R-base (∼300M parameters), and XLM-R-large (∼600M parameters). Systems for the first six language pairs in Table 2 were pre-trained on the parallel data while the system for Ru-En was only trained on the task data. We combined predictions on the first six language pairs as the submission to the multilingual task.

As shown in Table 1, unsupervised QE indicators obtained competitive results using sequence-level probability from NMT models. Disturbing the model parameters improves the performance of all language pairs. We did not combine the predictions from unsupervised methods into our submissions.

Table 2 lists the results of the system ensemble

System	Target	Source
RNN-word	0.467	-
Transformer-word	0.511	-
Transformer-subword	0.542	0.292
Deep Transformer-subword	0.545	-
Ensemble	0.610	0.308

Table 3: Results of the English-Chinese post-editing task. 'word' denotes the system uses word-level tokenization.

System	Target	Source
RNN-word	0.395	-
Transformer-word	0.413	-
Transformer-subword	0.451	0.285
Ensemble	0.500	0.347

Table 4: Results of the English-German post-editing tasks.

with pretraining and fine-tuning. We combined predictions from 10 pre-trained models with three different settings: mBERT, XLM-R-base, and XLM-R-large. We only report the results with the highest Pearson (r) correlation on the test data. We observe that larger models consistently outperformed small ones for all language pairs. Besides, ensemble methods significantly improved the performance on the test set. It also shows that the quality estimation of high-resource languages performs far worse than low-resource languages.

6.2 Task 2

For En-Zh, we trained 5-10 single models for each setting: token-based bi-RNNs (RNN-Token), token-based transformer (Trans-Token), BPE-based transformer (Trans-BPE), and BPE-based deep transformer with 25 encoder layers (Deep Trans). For En-De, we created three systems using the same architectures as En-Zh except for the deep transformer. We applied the multi-task learning strategies to the target-side word-level and sentence-level tasks described as §3.

Table 3 shows the results on the English-Chinese word-level task. Deep transformer and BPE tokenization bring the most gains to both the target-side MCC. Results on the English-German task are listed in Table 4. It shows that our ensemble methods are effective in boosting performance across different tasks.

System	F1 Score	Pearson
Transformer-word	0.373	-
Transformer-subword	0.400	-
Deep Transformer	0.402	-
mBERT	-	0.446
XLM-R	-	0.489
Ensemble	0.418	0.494

Table 5: Results of the document-level tasks. The deep transformer model contains 24 encoder layers and 6 decoder layers.

6.3 Task 3

Table 5 shows the results obtained by three different models and the ensemble on the annotation task. BPE brings about 0.03 points improvements of F1 scores on both the validation and test sets. The system ensemble further pushes the score by about 0.02. Table 5 also lists the results of the scoring task. We report the results of two pretraining methods and their ensemble on the test data. XLM-R outperformed the mBERT model by 0.04 points in the Pearson correlation, while the ensemble brought a slight benefit.

7 Conclusion

This paper describes the submissions of the NiuTrans Team to the WMT 2020 QE task. We explored the combination of transfer learning, multi-task learning, and model ensemble. Different level tasks show that deep transformer NMT models and multilingual pretraining methods significantly boost QE models' performance.

Although our system achieved impressive results in all tasks and language pairs, there are still many problems. For instance, the translation quality estimation of low-resource languages performs much better than that of high-resource. It raises the concern of whether our model learns the evaluation criteria instead of memorizing data, as suggested in Sun et al. (2020). Besides, strong NMT models help quality estimation, but can we use QE models to improve NMT systems' learning? We plan to answer these questions in the future and promote the joint improvement of QE and NMT models.

Acknowledgements

This work was supported in part by the National Science Foundation of China (Nos. 61876035 and 61732005) and the National Key R&D Program of

China (No.2019QY1801). The authors would like to thank anonymous reviewers for their comments.

References

John Blatz, Erin Fitzgerald, George Foster, Simona Gandrabur, Cyril Goutte, Alex Kulesza, Alberto Sanchis, and Nicola Ueffing. 2004. Confidence estimation for machine translation. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 315–321, Geneva, Switzerland. COLING.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, F. Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *ACL*.

J. Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.

Marina Fomicheva, Shuo Sun, Lisa Yankovskaya, Frédéric Blain, Francisco Guzmán, Mark Fishel, Nikolaos Aletras, Vishrav Chaudhary, and Lucia Specia. 2020. Unsupervised quality estimation for neural machine translation. *ArXiv*, abs/2005.10608.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks.

Chi Hu, Bei Li, Yinqiao Li, Ye Lin, Yanyang Li, Chenglong Wang, Tong Xiao, and Jingbo Zhu. 2020. The NiuTrans system for WNGT 2020 efficiency task. In *Proceedings of the Fourth Workshop on Neural Generation and Translation*, pages 204–210, Online. Association for Computational Linguistics.

Fabio Kepler, Jonay Trénous, Marcos Treviso, Miguel Vera, António Góis, M. Amin Farajian, António V. Lopes, and André F. T. Martins. 2019a. Unbabel's participation in the WMT19 translation quality estimation shared task. In *WMT*.

Fabio Kepler, Jonay Trénous, Marcos Treviso, Miguel Vera, and André F. T. Martins. 2019b. OpenKiwi: An open source framework for quality estimation. In *ACL*.

Hyun Kim, Jong-Hyeok Lee, and Seung-Hoon Na. 2017. Predictor-estimator using multilevel task learning with stack propagation for neural quality estimation. In *Proceedings of the Second Conference on Machine Translation*, pages 562–568, Copenhagen, Denmark. Association for Computational Linguistics.

Hyun Kim, Joon-Ho Lim, Hyun-Ki Kim, and Seung-Hoon Na. 2019. QE BERT: Bilingual BERT using multi-task learning for neural quality estimation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 85–89, Florence, Italy. Association for Computational Linguistics.

Bei Li, Yinqiao Li, Chen Xu, Y. Lin, Jiqiang Liu, H. Liu, Ziyang Wang, Y. Zhang, N. Xu, Zeyang Wang, Kai Feng, Hexuan Chen, Tengbo Liu, Yanyang Li, Qiang Wang, Tong Xiao, and Jingbo Zhu. 2019. The niutrans machine translation systems for wmt19. In *WMT*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *ACL*.

Lucia Specia, Frédéric Blain, Marina Fomicheva, Erick Fonseca, Vishrav Chaudhary, Francisco Guzmán, and André FT Martins. 2020. Findings of the wmt 2020 shared task on quality estimation. In *Proceedings of the Fifth Conference on Machine Translation: Shared Task Papers*.

Shuo Sun, Francisco Guzmán, and Lucia Specia. 2020. Are we estimating or guesstimating translation quality? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6262–6267, Online. Association for Computational Linguistics.

Marina Sánchez Torrón and Philipp Koehn. 2016. Machine translation quality and post-editor productivity. In *In Proceedings of AMTA*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, L. Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *ArXiv*, abs/1706.03762.

Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F. Wong, and Lidia S. Chao. 2019a. Learning deep transformer models for machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1810–1822, Florence, Italy. Association for Computational Linguistics.

Ziyang Wang, Hui Liu, Hexuan Chen, Kai Feng, Zeyang Wang, Bei Li, Chen Xu, Tong Xiao, and Jingbo Zhu. 2019b. Niutrans submission for ccmt19 quality estimation task. In *CCMT*.

Tong Xiao, Jingbo Zhu, Hao Zhang, and Qiang Li. 2012. Niutrans: An open source toolkit for phrase-based and syntax-based machine translation. In *ACL*.

Elizaveta Yankovskaya, Andre Tättar, and Mark Fishel. 2019. Quality estimation and translation metrics via pre-trained word and sentence embeddings. In *ACL*.

Two-Phase Cross-Lingual Language Model Fine-Tuning for Machine Translation Quality Estimation

Dongjun Lee
Bering Lab, Republic of Korea
djlee@beringlab.com

Abstract

In this paper, we describe the Bering Lab's submission to the WMT 2020 Shared Task on Quality Estimation (QE). For word-level and sentence-level translation quality estimation, we fine-tune XLM-RoBERTa, the state-of-the-art cross-lingual language model, with a few additional parameters. Model training consists of two phases. We first pre-train our model on a huge artificially generated QE dataset, and then we fine-tune the model with a human-labeled dataset. When evaluated on the WMT 2020 English-German QE test set, our systems achieve the best result on the target-side of word-level QE and the second best results on the source-side of word-level QE and sentence-level QE among all submissions.

1 Introduction

Machine translation quality estimation (QE) is the task of estimating the quality of machine-translated (MT) output given just the source text at various granularity levels (word, sentence, and document) (Fonseca et al., 2019). Word-level QE can be divided into target-side and source-side tasks. On the target-side, the goal is to predict whether each word in the MT sentence is `OK` or `BAD` and whether there are missing words between each word. The goal on the source-side is to predict whether each word in the source sentence is correctly translated or not. On the other hand, sentence-level QE aims to predict the Human Translation Error Rate (HTER) (Snover et al., 2006) of the MT sentence, which measures the required amount of human editing to fix the MT sentence.

In this paper, we propose a cross-lingual language model fine-tuning approach with a few additional parameters for word-level and sentence-level QE. As a pre-trained cross-lingual language model, we use XLM-RoBERTa (XLM-R) (Conneau et al., 2019), which shows state-of-the-art performance for a wide range of cross-lingual transfer tasks. In addition, since labeling the QE dataset requires a large amount of human labor, we generate and utilize a huge artificial QE dataset to improve the performance of our model. Our contributions are summarized as follows.

- We propose an XLM-R-based neural network architecture for the QE. Our model can be jointly trained for both word-level and sentence-level QE.
- We generate a huge artificial QE dataset based on a parallel corpus with OpenNMT-py (Klein et al., 2017) and the TER tool (Snover et al., 2006).
- We train our model in two phases. First, we train our model with a huge artificially generated dataset. Then, we fine-tune the model with a human-labeled dataset.

In the experiment using the WMT 2020 English-German word-level QE test set, we achieve an MCC of 0.597 and 0.454 for the target-side and source-side, respectively, showing the best and second best performance among all submitted systems, respectively. For the sentence-level QE test set, we achieve a Pearson correlation of 0.723, which ranks second among all submissions.

2 Methodology

We fine-tune XLM-R (Conneau et al., 2019) with a few additional parameters for sentence-level and word-level QE as described in Figure 1. We train our model in two phases: 1) pre-training with a huge artificial dataset and 2) fine-tuning with a human-labeled dataset.

2.1 Input Representation

We follow the tokenization and input representation methods of XLM-R. A source sentence and

Proceedings of the 5th Conference on Machine Translation (WMT), pages 1024–1028

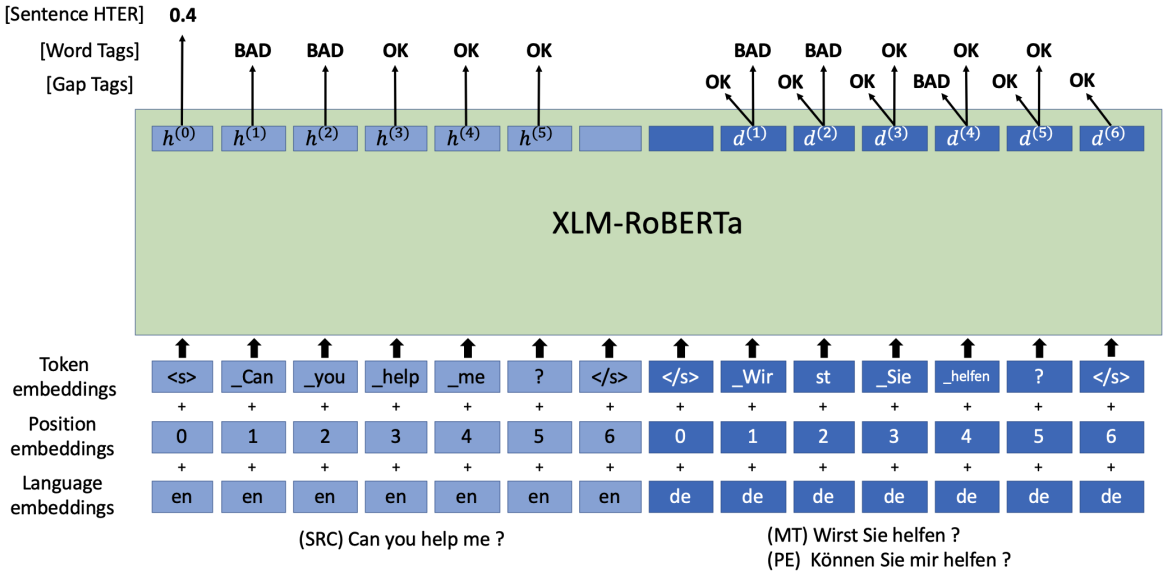


Figure 1: The XLM-R-based neural network architecture for word-level and sentence-level QE.

the corresponding MT sentence are tokenized with the same BPE model (Sennrich et al., 2016) that is trained based on shared vocabulary through languages. The input of the XLM-R model is a concatenated sequence of source tokens and MT tokens with special tokens (<s>, </s>) as follows:

<s> src_1, ..., $src_{|S|}$ </s> </s> mt_1, ..., $mt_{|T|}$ </s>

2.2 Sentence-level QE

For sentence-level QE, we use the final hidden vector $h^{(0)} \in \mathbb{R}^H$ of XML-R corresponding to the first input token (<s>) as the pooled representation. We use two linear layers with tanh activation to predict sentence-level HTER as follows:

$$r = W_s h^{(0)} + b_0 \quad (1)$$

$$y_{sent} = w_s^T tanh(r) + b_1 \quad (2)$$

where $W_s \in \mathbb{R}^{H \times H}$, $w_s \in \mathbb{R}^H$, $b_0 \in \mathbb{R}^H$ and $b_1 \in \mathbb{R}^1$ are trainable parameters and H is the dimension of hidden states.

The loss function L_{sent} is the mean squared error between y_{sent} and the true HTER $\hat{y}_{sent}$.

$$L_{sent} = MSE(y_{sent}, \hat{y}_{sent}) \quad (3)$$

2.3 Word-level QE

Word-level QE consists of two parts: the source-side and target-side. On the source-side, we predict whether each token in the source sentence is translated correctly or not. On the target-side, we predict whether each token in the MT sentence is OK or BAD, in addition to whether there are missing words between each word.

Source-side QE For source-side QE, we use the final hidden vector $h^{(i)} \in \mathbb{R}^H$ of XLM-R corresponding to each token in the source sentence. We introduce a linear layer and sigmoid activation to predict the probability that each token is BAD as follows:

$$P_{src}^{(i)} = sigmoid(w_o^T h^{(i)}), i \in (1, .., |S|) \quad (4)$$

where $w_o \in \mathbb{R}^H$ is a trainable parameter and $|S|$ is the number of tokens in the source sentence.

The loss function L_{src} is the binary cross entropy with an additional weight c for BAD examples as follows:

$$L_{src} = \frac{1}{|S|} \sum_{i=1}^{|S|} c\hat{y}_{src}^{(i)} \log P_{src}^{(i)} + (1 - \hat{y}_{src}^{(i)}) \log(1 - P_{src}^{(i)}) \quad (5)$$

Target-side QE For the target-side QE, we use the final hidden vector $d^{(i)} \in \mathbb{R}^H$ of XLM-R corresponding to each token in the MT sentence, including the last </s> token. We introduce two separate binary classification layers to predict the probability that each token in MT sentence is BAD as follows:

$$P_{tgt\_word}^{(i)} = sigmoid(w_w^T d^{(i)}), i \in (1, .., |T|) \quad (6)$$

and the probability that missing words exist before each token as follows:

$$P^{(i)}_{tgt\_gap} = sigmoid(w_g^T d^{(i)}), i \in (1, .., |T| + 1) \quad (7)$$

where $w_w, w_g \in \mathbb{R}^H$ are trainable parameters and $|T|$ is the number of tokens in the machine translated sentence.

The loss function for target-side QE L_{tgt} is the sum of the binary cross entropy for word L_{tgt_word} and gap L_{tgt_gap} that are defined in the same manner as Eq. (5).

$$L_{tgt} = L_{tgt\_word} + L_{tgt\_gap} \quad (8)$$

2.4 Pre-Training on Artificial Dataset

Building the Artificial Dataset Labeling data for QE requires the triplets of source sentences, machine-translated (MT) sentences, and human post-edited (PE) sentences. Since huge costs are required to achieve PE sentences, we use a parallel corpus that includes only source sentences and target sentences to build artificial triplets following the ideas from Negri et al. (2018).

First, we split the parallel corpus into a training set and test set. We train an NMT model with the training set and use the test set to generate artificial triplets. We generate MT sentences based on the trained NMT model and we use the target sentences of the parallel corpus as PE sentences. We repeat this process with different data splits to build huge artificial triplets. Finally, we use the TER tool[1] (Snover et al., 2006) to annotate sentence-level HTER scores and word-level tags for the MT sentences. We do not annotate source-side word-level tags in this work as it additionally requires word alignment between source sentences and MT sentences.

Pre-training QE Model We first pre-train our QE model with only the artificial dataset. In the pre-training step, we jointly train sentence-level QE and target-side word-level QE on a single model. The loss function for the pre-training step L_{pre_train} is the sum of the loss for sentence-level QE and target-side word-level QE.

$$L_{pre\_train} = L_{sent} + L_{tgt} \quad (9)$$

Since our artificial dataset does not include source-side word-level tags, we do not include the training objective for source-side word-level QE in the pre-training step.

[1] http://www.cs.umd.edu/~snover/tercom/

2.5 Fine-Tuning on Human-Labeled Dataset

After the pre-training, we fine-tune the model with only a human-labeled dataset. Unlike the pre-training step, each QE model (sentence-level, source-side and target-side of word-level) is trained separately in the fine-tuning step.

For the sentence-level and target-side of word-level QE models, all the parameters are initialized with trained weights from the pre-training step. However, since our pre-trained model does not include source-side word-level QE, we randomly initialize the weight of a source-side specific parameter (w_o in Eq. (4)) and the rest of the parameters are initialized with weights from the pre-trained model.

2.6 Ensemble

For the sentence-level ensemble, we average the HTER prediction of multiple models. For the word-level, we use the majority voting ensemble.

3 Experiments

3.1 Experimental Setup

We evaluate our model with WMT 2020 English-German QE dataset.[2] For the sentence-level QE evaluation, we use the Pearson correlation for sentence-level HTER prediction. For the word-level QE evaluation, we use the Matthews correlation coefficient (MCC) for both the target-side and source-side.

To generate an artificial dataset for pre-training (§2.4), we use the English-German parallel corpus provided by the shared task that consists of 23,440,059 pairs. We use 90% of the pairs to train a Transformer-based (Vaswani et al., 2017) NMT model with OpenNMT-py (Klein et al., 2017) and the rest of the pairs are used to generate artificial triplets. As a result of running the process five times with different data splits, we achieve 11,720,029 artificial triplets.

For the fine-tuning, we use only the official QE dataset that consists of 7,000 triplets as a human-labeled dataset.

3.2 Model Configuration

We use XLM-R-Large (Conneau et al., 2019) as a pre-trained cross-lingual language model. For pre-training with the artificial dataset, we use the Adam optimizer (Kingma and Ba, 2014) with a

[2] http://www.statmt.org/wmt20/quality-estimation-task.html

Systems	Pearson↑	Target-side MCC↑	Source-side MCC↑
Ours	**0.715**	**0.591**	**0.464**
-ensemble	0.712	0.586	0.457
-ensemble -pre-train	0.591	0.476	0.365
-ensemble -fine-tune	0.424	0.378	-

Table 1: Ablation analysis for sentence-level and word-level QE on the WMT 2020 English-German QE *dev* set. Since our pre-training step does not include source-side word-level QE, we do not measure the source-side MCC for the pre-trained only model.

Systems	Pearson↑	MAE↓	RMSE↓
HW-TSC	**0.758**	**0.099**	**0.133**
Ours (Bering Lab)	0.723	0.107	0.140
NiuTrans	0.649	0.123	0.154
IST and Unbabel	0.633	0.137	0.178
NJUNLP	0.618	0.129	0.160
Baseline	0.392	0.150	0.190

Table 2: Top-5 and baseline systems from the official result for the sentence-level QE on the WMT 2020 English-German QE shared task.

Systems	Target-side MCC↑	Source-side MCC↑
Ours (Bering Lab)	**0.597**	0.454
HW-TSC	0.583	**0.523**
NiuTrans	0.500	0.347
NICT Kyoto	0.485	0.353
IST and Unbabel	0.465	0.349
Baseline	0.358	0.266

Table 3: Top-5 and baseline systems from the official result for the word-level QE on the WMT 2020 English-German QE shared task.

learning rate of 5e-6, and a batch size of 8 for 2 epochs. Additionally, we use dropout (Hinton et al., 2012) with a rate of 0.1 for the regularization. For word-level QE, we use a weight of 3.0 on the `BAD` class (c). For fine-tuning with the human-labeled dataset, we follow the same hyperparameters as the pre-training step but for 5 epochs with early stopping. For the ensembling, we train five models with different random seeds.

3.3 Experimental Result

Table 1 shows the result of ablation analysis for sentence-level and word-level QE on the *dev* set. We conduct an ablation analysis of three aspects: 1) without an ensemble, 2) without pre-training with artificially generated dataset, 3) without fine-tuning with human-labeled dataset. When our model is trained with only the human-labeled dataset, Pearson correlation, target-side MCC and source-side MCC drop by 0.12, 0.11, and 0.09, respectively. This result demonstrates that pre-training with the artificial dataset significantly improves performance for both sentence-level and word-level QE. When our model is trained with only the artificial dataset, Pearson correlation and target side MCC drop by 0.29 and 0.21, respectively. This result shows that fine-tuning with a human-labeled dataset is essential for our performance.

Table 2 and 3 shows the official results for sentence-level and word-level QE for the WMT 2020 QE shared task. For both sentence-level and word-level QE, our systems significantly outperformed the official baseline systems (Kepler et al., 2019). Moreover, we achieve the best result on the target side of word-level QE among all submitted systems. We also achieve the second best

results on the source side of word-level QE and sentence-level QE.

4 Conclusion

This paper describes Bering Lab's submissions to the WMT 2020 QE shared task. We propose a two-phase cross-lingual language model fine-tuning approach for word-level and sentence-level translation quality estimation. The experimental results show that pre-training with an artificially generated dataset significantly improves performance for both tasks. Overall, our submitted systems achieve the best result on the target side of word-level QE and the second best results on the source side of word-level QE and the sentence-level QE among all submissions.

References

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Erick Fonseca, Lisa Yankovskaya, André FT Martins, Mark Fishel, and Christian Federmann. 2019. Findings of the wmt 2019 shared tasks on quality estimation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 1–10.

Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.

Fabio Kepler, Jonay Trénous, Marcos Treviso, Miguel Vera, and André FT Martins. 2019. Openkiwi: An open source framework for quality estimation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 117–122.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proc. ACL*.

Matteo Negri, Marco Turchi, Rajen Chatterjee, and Nicola Bertoldi. 2018. Escape: a large-scale synthetic corpus for automatic post-editing. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of association for machine translation in the Americas*, volume 200.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

IST-Unbabel Participation in the WMT20 Quality Estimation Shared Task

João Moura
Instituto Superior Técnico, Lisbon
joaopcmoura@tecnico.ulisboa.pt

Miguel Vera
Unbabel, Lisbon
miguel.vera@unbabel.com

Daan van Stigt
Unbabel, Lisbon
daan.stigt@unbabel.com

Fabio Kepler
Unbabel, Lisbon
kepler@unbabel.com

André F. T. Martins
Instituto de Telecomunicações
Instituto Superior Técnico
Unbabel, Lisbon
andre.t.martins@tecnico.ulisboa.pt

Abstract

We present the joint contribution of IST and Unbabel to the WMT 2020 Shared Task on Quality Estimation. Our team participated on all tracks (Direct Assessment, Post-Editing Effort, Document-Level), encompassing a total of 14 submissions. Our submitted systems were developed by extending the OpenKiwi framework to a transformer-based predictor-estimator architecture, and to cope with glass-box, uncertainty-based features coming from neural machine translation systems.

1 Introduction

Quality estimation (QE) is the task of evaluating a translation system's quality without access to reference translations (Blatz et al., 2004; Specia et al., 2018). This paper describes the joint contribution for Instituto Superior Técnico (IST) and Unbabel to the WMT20 Quality Estimation shared task, where systems were submitted to all three tasks: 1) sentence-level direct assessment; 2) word and sentence-level post-editing effort; and 3) document-level annotation and scoring.

Unbabel's participation in previous editions of the shared task (2016, 2017, 2019) used ensemble of strong individual systems, with varying architectures and hyper-parameters. While this strategy led to very strong results, large system ensembles are not a very practical solution, complicating model deployment and requiring expensive computation and memory usage. This year, in contrast, our focus was on simplicity: only single model systems were submitted and, in a few cases, an additional simple ensemble of the same model. Transfer learning on top of pretrained multilingual models was also used for avoiding manual pretraining for each language pair.

Last year's winning submission (Kepler et al., 2019a) combined strong individual systems built on top of the OpenKiwi framework (Kepler et al., 2019b) and pretrained Transformer models. We consolidated those changes with support for newly released pretrained models and packages and published a new version 2.0 of the OpenKiwi framework.[1] We trained and submitted single model systems in OpenKiwi for all tasks, beating all baselines by a large margin. Additionaly, we also used OpenKiwi with small adaptations to handle specific sources of information in Tasks 1 and 3.

Task 1, in particular, was introduced this year with Direct Assessment scores as targets. Further, it introduced the novelty of providing the trained NMT models that were used for producing the translations. Previously, only black-box QE was considered in the WMT Shared Task, as it is one of the main uses cases. With the availability of the NMT models, new glass-box approaches can be explored. Our best submitted systems drew inspiration from (Fomicheva et al., 2020) to leverage this information, improving in performance and robustness over a black-box approach.

Our main contributions are:

- We release the second version of OpenKiwi along with our submission, with a variety of new features, including the ability to use pretrained Transformer-based Language Models;
- We show that transfer learning techniques still perform well, by fine-tuning *XLM-Roberta* in a Predictor-Estimator architecture;
- We incorporate features extracted from the provided NMT models into our existing architectures and show that glass-box QE improves upon black-box approaches.

[1] The new version will be publicly available at https://github.com/unbabel/openkiwi.

Proceedings of the 5th Conference on Machine Translation (WMT), pages 1029–1036

2 Quality Estimation Tasks

This year's shared task edition comprised three tasks: 1) a newly introduced one for sentence-level direct assessment; 2) one for word and sentence-level post-editing effort; and 3) one for document-level. Refer to the Findings paper (Specia et al., 2020) for full descriptions.

Of noteworthy mention is that the NMT models for Tasks 1 and 2 where provided along with the data, which opened up the possibility of using glass-box approaches.

3 Implemented Systems

To avoid the complexity of ensemble of several systems, all our submitted systems consisted of a single model type. In addition to standard OpenKiwi 2.0 systems submitted to Tasks 1 and 2 (§3.1), we implemented two types of extensions on top of OpenKiwi, one for exploring glass-box approaches for Tasks 1 and 2 (§3.2), and one for handling document-level QE for Task 3 (§3.3).

3.1 Base OpenKiwi System

Given the success in doing transfer learning with pretrained Language Models in last year's shared task edition, we published support for them as part of the open source QE framework OpenKiwi in a new 2.0 version. BERT, XLM, and XLM-Roberta are currently supported via the `Transformers`[2] Python package (Wolf et al., 2019), which means different models can be easily used. For this year's shared task, we based all systems on this version of OpenKiwi and used pretrained XLM-Roberta models (Conneau et al., 2020), either `base` or `large` versions. We chose XLM-Roberta (called XLM-R from here on) instead of XLM, used in last year's best individual model, due to its reported state-of-the-art performance on downstream cross-lingual tasks and based on preliminary experiments.

The architecture follows the overall pattern introduced originally in the Predictor-Estimator model (Kim et al., 2017), comprising a "Feature Extractor" module with a "Quality Estimator" module on top. Figure 1 depicts this general architecture.

The Feature Extractor module consists of a pretrained XLM-R model and feature extraction methods on top, such that features for the target sentence, the target tokens, and the source tokens are returned separately. Source and target sentences

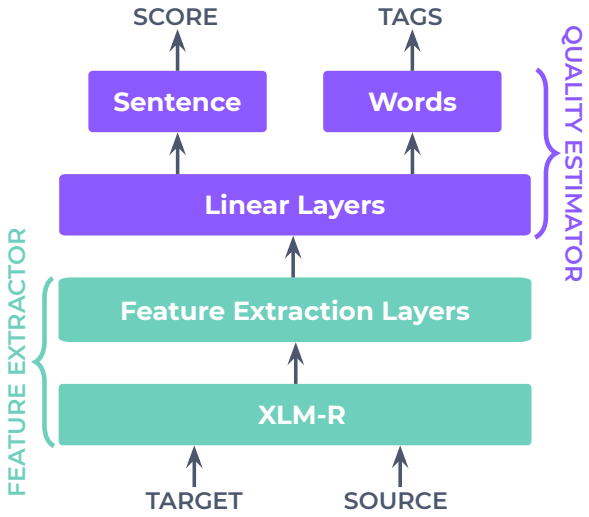


Figure 1: General architecture of the implemented OpenKiwi-based systems.

are passed as inputs in the format `<s> target </s> <s> source </s>`. Output features for tokens in the target sentence are averaged and then concatenated with the classifier token embedding (first `<s>` in the input), and returned as sentence features.[3]

For the Quality Estimator module we used linear layers instead of a bi-LSTM (as used by Kim et al. (2017)), since initial experiments showed similar performance. Additional linear layers were stacked on top for each output type: target words, target gaps, source words, and sentence regression.

For the plain OpenKiwi submissions we used the XLM-R `base` model and a Quality Estimator block with two linear layers. Hyper-parameter search was performed for each language pair and task[4] and submitted as a single model system to Tasks 1 and 2, and used as basis for the submission to Task 3. These systems will be referred to as OpenKiwi-base through the rest of the paper.

[2] https://github.com/huggingface/transformers

[3] Even though XLM-R was not trained on the Next Sentence Prediction objective (therefore not using the classification token in its original pretraining), preliminary experiments showed that concatenating inputs, average pooling, and using the classification token resulted in better performance compared to feeding source and target separately and extracting sentence features with other strategies (only pooled target, only the classifier token, classifier token + pooled source, and others).

[4] Hyper-parameters that were searched are: learning rate, dropout, number of warmup steps, and number of freeze steps.

3.2 Glass-Box QE

3.2.1 Glass-Box Features

Recent work on MT confidence estimation (Fomicheva et al., 2020) showed that useful information coming from an MT system, obtained as a by-product of translation, can be competitive with supervised black-box QE models in terms of correlation to human judgements of translation quality, in settings where the labeled data is scarce. The approach described in Fomicheva et al. (2020) requires access to the MT system that produced the translations (unlike the black-box regime). This year's new Task 1, and the fact it shares datasets with Task 2, allowed us to explore this approach on both tasks. In our work, we investigated how to combine the richness of this extra information coming from the provided Neural MT (NMT) system with the strength of state-of-the-art approaches to supervised QE.

To this end, we extract features (referred to as *glass-box features* henceforth) using the output probability distribution obtained from (i) a standard deterministic NMT and (ii) using uncertainty quantification. For (ii) we use Monte Carlo Dropout (Gal and Ghahramani, 2015) as a way of circumventing the miscalibration problem of Deep Neural Networks (Guo et al., 2017) and obtaining measures indicative of the model's uncertainty.

We obtain 7 different features for each sentence of each language-pair, the first 3 via (i) and the last 4 via (ii) (full details are in Fomicheva et al. (2020)):

- `TP` - sentence average of word translation probability
- `Softmax-Ent` - sentence average of softmax output distribution entropy
- `Sent-Std` - sentence standard deviation of word probabilities
- `D-TP` - average `TP` across $N (N = 30)$ stochastic forward-passes
- `D-Var` - variance of `TP` across N stochastic forward-passes
- `D-Combo` - combination of `D-TP` and `D-Var` defined by $1 - \texttt{D-TP}/\texttt{D-Var}$
- `D-Lex-Sim` - lexical similarity - measured by `METEOR` score (Banerjee and Lavie, 2005) - of MT output generated in different stochastic passes.

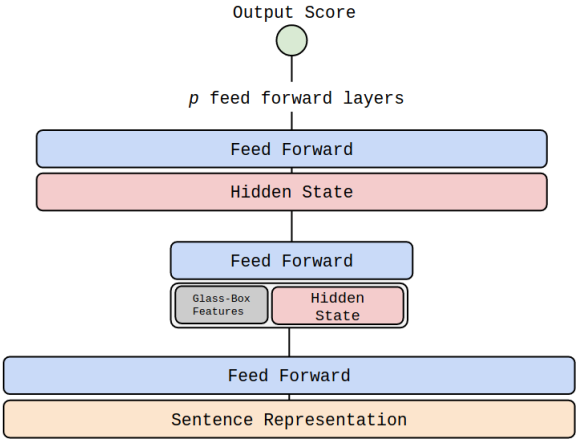


Figure 2: Architecture of the "Quality Estimator" module modified to include *glass-box features*.

Table 1 shows the correlation between each one of these features and human DAs for every language pair in Task 1. As expected, features obtained using uncertainty quantification consistently display higher correlations across all language-pairs, `D-TP` being the most effective for high and medium resource languages, and `D-Lex-Sim` for low resource languages.

3.2.2 Glass-box + Black-box Model

Different configurations were attempted in order to introduce the extracted glass-box features into the OpenKiwi system. The best empirical performance was observed with a simple method: we reduced the dimension of the pooled sentence features output from XLM-R by about five fold (onto `bottleneck_size`), creating a dimensional bottleneck and forcing a more compact sentence representation, and then concatenated the seven extracted glass-box features to this hidden state, followed by an expansion back to a higher dimensional state of `hidden_size`. The result is used as input feature for regression on the sentence score, employing `p` progressively smaller feed-forward layers (halving in size). A visualization of this process can be seen in Figure 2.

The glass-box features were individually normalized a priori, according to their mean and variance in the training dataset, allowing for their integration in the network's training in a scale-independent way.

Systems were trained for all language pairs in Tasks 1 and 2. XLM-R `large` was used instead of `base` version. We ran experiments with and without glass-box features. From here on we will

	Feature	Language Pair						
		En-De	En-Zh	Ro-En	Et-En	Ne-En	Si-En	Ru-En
(i)	TP	0.0993	0.2808	0.5951	0.3992	0.3653	0.3658	0.3658
	Softmax-Ent	0.0858	0.2919	0.5595	0.3546	0.4133	0.4077	0.3790
	Sent-Std	0.0691	**0.3252**	0.5049	0.3985	0.3669	0.3912	0.3510
(ii)	D-TP	**0.1078**	0.3158	**0.6404**	**0.4936**	0.3905	0.3797	**0.4441**
	D-Var	0.0782	0.1943	0.3550	0.2780	0.2336	0.2338	0.2329
	D-Combo	0.0487	0.1259	0.2620	0.1335	0.2938	0.2244	0.2013
	D-Lex-Sim	0.0994	0.2903	0.6210	0.3940	**0.4751**	**0.4318**	0.4092

Table 1: Pearson correlation (r) between the employed *glass-box features* and human DA's for every language pair in Task 1 (validation set) - best results are in bold.

call KIWI-GLASS-BOX the system as described here, which was the one used for the official submissions, but for comparison we will also refer to KIWI-LARGE as the same system but without using the glass-box features.

Hyper-parameter search was performed over `p`, `bottleneck_size`, `hidden_size`, `warmup_steps` (number of warm up steps for optimizer), `freeze_steps` (number of steps for which XLM-R's weights are not updated) and `lr` (learning rate). The exact values can be found in Table 6 in Appendix A.

All submissions of KIWI-GLASS-BOX to Task 1 were created by simple linear ensembles, combining 5 of the models obtained through hyper-parameter search for each language pair. We used the validation set predictions of these 5 models to train a LASSO regression model. However since we do not possess labels for the test set, these ensembles were trained using k-fold cross-validation ($k = 10$) on the validation set.

3.3 Document-level QE

For Task 3 we submitted two systems, both of which are based on the general OpenKiwi architecture described in Section 3.1. The two systems differ only in the type of tags they predict, and the subsequent post-processing that is applied to these tags to obtain annotations and document-level MQM (Multidimensional Quality Metrics) scores. We submitted single systems that predict both tasks of document-level annotation and scoring.

The first system, henceforth referred to as KIWI-DOC, is OPENKIWI-BASE with additional data processing to convert between word- and sentence-level predictions, and document-level predictions. The data approach is the exact same as Kepler et al. (2019a). To obtain training data, annotations are converted to binary word-level tags (`OK` and `BAD` tags) and sentence-level MQM scores are computed from the annotations pertaining to the sentence. After training, document-level annotation predictions are obtained by the following heuristic: contiguous `BAD` tags in the word-level predictions are grouped into a single annotation span and are given the severity label `major`. Predicted document-level MQM scores are obtained by averaging predicted sentence-level MQM weighted by sentence-length (`regression`) or by direct computation from the predicted annotations using the MQM formula (`direct`).

The second system, KIWI-DOC-IOB, is a new contribution in which the task of annotating is approached as Named-entity recognition by using severity tags in IOB (Inside-Outside-Beginning) format.[5] This richer tag scheme addresses two types of information loss that occur in the approach taken for KIWI-DOC: the severity information is kept, and adjacent but disjoint annotations are not collapsed into single annotations during prediction.[6] This approach has the advantage that the predicted tag sequences can be converted to annotations directly by converting the token spans into character spans and using the predicted label as severity. The architecture of KIWI-DOC-IOB is identical to that of KIWI-DOC except that it is trained with a linear chain CRF[7] that enforces correctness of the IOB tag-sequence at prediction time[8].

For both systems we trained a final linear regression model that combines the two types of pre-

[5]The full label set is hence: `B-minor`, `I-minor`, `B-major`, `I-major`, `B-critical`, `I-critical`, and `O`.

[6]The two other types of information loss that were noted by Kepler et al. (2019a) are left unaddressed: tags are still defined at the token-level, and annotations consisting of multiple spans are still split into individual annotations.

[7]Each edge score is a single learned parameter that is independent of the input.

[8]During decoding, the edge scores corresponding to the impossible transitions are set manually to $-\infty$.

dicted MQM scores (`regression` and `direct`) with features derived from the tag-level predictions. We use the following additional features (when available[9]) computed over the document: the fraction of predicted tags corresponding to an error tag;[10] and the mean, variance, minimum, and maximum of the probability of the `BAD`. For simplicity we train the linear regression on the same training data as the systems. For each system, we perform search over all combinations of features, and choose the subset that gives the highest Pearson score on the validation set for that particular system.

4 Experimental Results

4.1 Task 1: Sentence-Level Direct Assessment

The results achieved over the validation set on all language pairs for Task 1 are shown in Table 2. We also include the best correlation achieved by any *glass-box feature* (denoted by BEST GB FEATURE), showing that indeed the proposed method allows for this rich information to complement and enhance the model's training, resulting in a performance increase when compared to model or GB-feature independently.

High resource language pair models (*En-De, En-Zh, Ru-En* benefit the most from the aid of NMT internal information, in particular English-German, where an increase of $\approx 4.5\%$ occurs; this might indicate the usefulness of incorporating nuanced information when sentence scores have less variability.

Scored test set predictions submitted during the development of this approach served as informative feedback, revealing the drop from validation to test performance to be smaller on KIWI-GLASS-BOX models when compared to KIWI-LARGE models, suggesting better generalization capabilities.

4.2 Task 2: Word and Sentence-Level Post-editing Effort

We trained OPENKIWI-BASE and KIWI-GLASS-BOX on all three subtasks at the same time: source tags, target tags, and sentence HTER. The best model was selected by the highest sum of the three metrics on the validation set. We used a single run of each of the two models to simultaneously predict the three outputs. The results can be seen in Table 3. Using the glass-box features provided a significant boost to the Pearson score, showing our strategy for sentence-level DA estimation performed well also when estimating sentence-level HTER.

Even though we only have a single model for all subtasks, our models outperformed the baselines by a large margin and performed very competitively in the test leaderboard (to cite Findings paper).

Pair	System	Pearson VAL	Pearson TEST
En-De	(*)**KIWI-GLASS-BOX-ENSEMBLE**	0.5715	0.5230
	KIWI-GLASS-BOX	0.5263	-
	KIWI-LARGE	0.4794	-
	OPENKIWI-BASE	0.3499	0.2670
	BEST GB FEATURE	0.1078	-
	Openkiwi 1.0	-	0.1455
En-Zh	(*)**KIWI-GLASS-BOX-ENSEMBLE**	0.5711	0.4940
	KIWI-GLASS-BOX	0.5461	-
	KIWI-LARGE	0.5258	-
	OPENKIWI-BASE	0.4199	0.3460
	BEST GB FEATURE	0.3252	-
	OpenKiwi 1.0	-	0.1902
Ro-En	(*)**KIWI-GLASS-BOX-ENSEMBLE**	0.8968	0.8910
	KIWI-GLASS-BOX	0.8841	-
	KIWI-LARGE	0.8790	-
	OPENKIWI-BASE	0.6672	0.7080
	BEST GB FEATURE	0.6404	-
	OpenKiwi 1.0	-	0.6845
Et-En	(*)**KIWI-GLASS-BOX-ENSEMBLE**	0.7697	0.7700
	KIWI-GLASS-BOX	0.7611	-
	KIWI-LARGE	0.7496	-
	OPENKIWI-BASE	0.6728	0.6900
	BEST GB FEATURE	0.4936	-
	OpenKiwi 1.0	-	0.4770
Ne-En	(*)**KIWI-GLASS-BOX-ENSEMBLE**	0.7994	0.7920
	KIWI-GLASS-BOX	0.7804	-
	KIWI-LARGE	0.7711	-
	OPENKIWI-BASE	0.6987	0.6040
	BEST GB FEATURE	0.4751	-
	OpenKiwi 1.0	-	0.3860
Si-En	(*)**KIWI-GLASS-BOX-ENSEMBLE**	0.6896	0.6390
	KIWI-GLASS-BOX	0.6604	-
	KIWI-LARGE	0.6521	-
	OPENKIWI-BASE	0.5727	0.5650
	BEST GB FEATURE	0.4318	-
	OpenKiwi 1.0	-	0.3737
Ru-En	(*)**KIWI-GLASS-BOX-ENSEMBLE**	0.7391	0.7670
	KIWI-GLASS-BOX	0.7137	-
	KIWI-LARGE	0.6938	-
	OPENKIWI-BASE	-	-
	BEST GB FEATURE	0.4441	-
	OpenKiwi 1.0	-	0.5479

Table 2: Task 1 results on the validation and test sets for all language pairs in terms of Pearson's r correlation. Systems in **bold** were officially submitted. (*) Lines with an asterisk use LASSO regression to tune ensemble weights on the validation set, therefore their numbers cannot be directly compared to the other models.

4.3 Task 3: Document-Level QE

The results for the document-level scoring are shown in Table 4. For both systems we observe

[9] Because of the non-binary tags and CRF model the probability based features are not used for the KIWI-DOC-IOB model (posterior marginals could be used for this).

[10] This correspond to the `BAD` tag for KIWI-DOC and all tags different from `O` for KIWI-DOC-IOB.

Pair	System	Target MCC		Source MCC		Pearson	
		Val	Test	Val	Test	Val	Test
En-De	KIWI-GLASS-BOX	**0.460**	**0.465**	**0.357**	**0.349**	**0.618**	**0.633**
	OPENKIWI-BASE	0.445	0.432	0.330	0.324	0.561	0.531
	(*)OpenKiwi 1.0	-	0.358	-	0.266	-	0.392
En-Zh	KIWI-GLASS-BOX	0.567	0.567	**0.348**	0.287	**0.691**	**0.651**
	OPENKIWI-BASE	**0.576**	**0.575**	0.298	0.287	0.615	0.593
	(*)OpenKiwi 1.0	-	0.509	-	0.270	-	0.506

Table 3: Task 2 word and sentence-level results on the validation and test sets. Results for OPENKIWI-BASE and KIWI-GLASS-BOX were obtained from a single model trained by multi-tasking on the 3 different subtasks. (*) Baseline results on the validation set were not made available by the organizers.

System	Validation	Test
KIWI-DOC-`regression`	0.5146	0.4127
KIWI-DOC-`direct`	0.3131	0.3156
KIWI-DOC-`linear`	0.5635	0.4014
KIWI-DOC-IOB-`regression`	0.5731	**0.4746**
KIWI-DOC-IOB-`direct`	0.5483	0.3363
KIWI-DOC-IOB-`linear`	**0.6023**	0.4493

Table 4: Results of document-level (task 3) submissions for MQM scoring (Pearson). The results of KIWI-DOC and KIWI-DOC-IOB are for the same single model. For model selection during training we used the summed validation set Pearson of `direct` and `regression` to obtain a model that performs well in both methods.

System	Validation	Test
KIWI-DOC	**0.4934**	**0.4716**
KIWI-DOC-IOB	0.4016	0.4147

Table 5: Results of document-level (task 3) submissions for annotation (F1). For model selection during training we used validation set MCC for KIWI-DOC and validation set tagging F1 for KIWI-DOC-IOB.

a large drop in Pearson score from validation set to test set, in the range of 0.1-0.2,[11] which suggests that there is a difference in data distribution between the two sets. On the validation set, KIWI-DOC and KIWI-DOC-IOB obtain comparable Pearson correlation, albeit for different MQM methods. While both models perform comparably in the sentence score prediction (`regression`), the KIWI-DOC-IOB system clearly outperforms KIWI-DOC on the MQM scores that are computed directly from the predicted annotations (`direct`). The improvements made by linear regression on the validation set do not consistently translate to the test set. This suggests that our method of search over features for the linear regression is overly optimizing the performance to the validation data. It may also reflect our choice to train the linear model on system predictions on training data.

Table 5 shows the results for the annotation task. The best results are obtained by KIWI-DOC. Surprisingly, the strong scoring results of KIWI-DOC-IOB with `direct` (derived from predicted annotations) do not translate to good results on the annotation F1. The difference between the models is caused by the different trade-off between precision and recall: KIWI-DOC-IOB produces less annotations that are more precise, but KIWI-DOC catches much more errors.[12] The most likely cause for this is the more complex tag-set and constrained decoding of KIWI-DOC-IOB.

5 Conclusions

Our approach to this year's edition of the QE shared task was simplicity. Our submissions consisted of either single models, or simple ensembles of multiple runs of the same model. Moreover, we used multi-task models in Task 2, where a system was trained on all three possible outputs (target and source word level and sentence level). We implemented a new version of OpenKiwi and used it as our baseline. It significantly outperformed the official shared task baseline across the board, which was based on the previous version of OpenKiwi. Finally, we showed that having access to NMT models enables using glass-box approaches to QE, which in turn improves performance when used in

[11] The only exception is KIWI-DOC-IOB-`direct`, which performanced equally poorly on both.

[12] On the validation set KIWI-DOC-IOB predicted 2555 annotations, whereas KIWI-DOC predicted 4028 (the gold set has 5626 annotations). Extending the output message of the annotation evaluation script allowed us to further validate this hypothesis on the validation set: for KIWI-DOC-IOB precision/recall is 0.6287/0.3322; for KIWI-DOC precision/recall is 0.4549/0.6092.

combination with a black-box QE system.

Acknowledgments

This work was supported by the P2020 programs MAIA (contract 045909) and Unbabel4EU (contract 042671), by the European Research Council (ERC StG DeepSPIN 758969), and by the Fundação para a Ciência e Tecnologia through contract UID/50008/2019.

References

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

John Blatz, Erin Fitzgerald, George Foster, Simona Gandrabur, Cyril Goutte, Alex Kulesza, Alberto Sanchis, and Nicola Ueffing. 2004. Confidence Estimation for Machine Translation. In *Proc. of the International Conference on Computational Linguistics*, page 315.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Marina Fomicheva, Shuo Sun, Lisa Yankovskaya, Frédéric Blain, Francisco Guzmán, Mark Fishel, Nikolaos Aletras, Vishrav Chaudhary, and Lucia Specia. 2020. Unsupervised quality estimation for neural machine translation. *Transactions of the Association for Computation Linguistics*.

Yarin Gal and Zoubin Ghahramani. 2015. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. *Proceedings of The 33rd International Conference on Machine Learning*.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On calibration of modern neural networks. *ArXiv*, abs/1706.04599.

Fabio Kepler, Jonay Trénous, Marcos Treviso, Miguel Vera, António Góis, M. Amin Farajian, António V. Lopes, and André F. T. Martins. 2019a. Unbabel's participation in the WMT19 translation quality estimation shared task. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 78–84, Florence, Italy. Association for Computational Linguistics.

Fabio Kepler, Jonay Trénous, Marcos Treviso, Miguel Vera, and André F. T. Martins. 2019b. OpenKiwi: An open source framework for quality estimation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 117–122, Florence, Italy. Association for Computational Linguistics.

Hyun Kim, Jong-Hyeok Lee, and Seung-Hoon Na. 2017. Predictor-Estimator using Multilevel Task Learning with Stack Propagation for Neural Quality Estimation. In *Conference on Machine Translation (WMT)*.

Lucia Specia, Frederic Blain, Marina Fomicheva, Erick Fonseca, Vishrav Chaudhary, Francisco Guzman, and Andre FT Martins. 2020. Findings of the wmt 2020 shared task on quality estimation. In *Proceedings of the Fifth Conference on Machine Translation: Shared Task Papers*.

Lucia Specia, Carolina Scarton, and Gustavo Henrique Paetzold. 2018. Quality Estimation for Machine Translation. *Synthesis Lectures on Human Language Technologies*, 11(1):1–162.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

A Hyper-parameters

Table 6 shows the hyperparameters used in Task 1.

Language Pair	Hyper-parameters				
	hidden_size	bottleneck_size	lr	warmup_steps	freeze_steps
EN-DE	900	200	1.00E-05	6535	750
EN-ZH	700	300	7.00E-06	3280	4375
RO-EN	900	200	9.00E-06	2625	5687
ET-EN	500	200	7.00E-06	655	3935
NE-EN	900	200	1.20E-05	2625	3060
SI-EN	900	200	7.00E-06	5250	5250
RU-EN	700	200	1.70E-05	3800	6125

Table 6: Hyper-parameters of the best models trained for each language pair in Task 1. 70 trials were performed for each search, using the OPTUNA framework (Akiba et al., 2019), and hyper-parameter values were sampled with the TPE (Tree-structured Parzen Estimator) algorithm. The criterion for trial selection was r Pearson correlation to validation set DA's.

TMUOU Submission for WMT20 Quality Estimation Shared Task

Akifumi Nakamachi
Osaka University
nakamachi.akifumi@ist.osaka-u.ac.jp

Hiroki Shimanaka
Tokyo Metropolitan University
shimanaka-hiroki@ed.tmu.ac.jp

Tomoyuki Kajiwara
Osaka University
kajiwara@ids.osaka-u.ac.jp

Mamoru Komachi
Tokyo Metropolitan University
komachi@tmu.ac.jp

Abstract

We introduce the TMUOU[1] submission for the WMT20 Quality Estimation Shared Task 1: Sentence-Level Direct Assessment. Our system is an ensemble model of four regression models based on XLM-RoBERTa with language tags. We ranked 4th in Pearson and 2nd in MAE and RMSE on a multilingual track.

1 Introduction

Quality Estimation (QE) is a task of estimating translation quality without reference sentences (Gandrabur and Foster, 2003; Blatz et al., 2004; Specia et al., 2018). Automatic evaluation metrics based on reference sentences, such as BLEU (Papineni et al., 2002), have contributed to improving translation quality on benchmark datasets. However, in situations where machine translation (MT) is actually used, these metrics are sometimes unable to assess the translation quality owing to the lack of reference sentences. The development of QE methods that are well correlated with manual evaluations enable users to decide whether to use the translation results as is, post-edit the results, or employ other machine translations.

At the Conference on Machine Translation (WMT), there have been conducted several QE-related competitions such as the QE task (Fonseca et al., 2019) for estimating post-edit rate HTER (Snover et al., 2006) and the QE as a Metric task (Ma et al., 2019) for relative evaluations of translation quality. This year, the WMT QE task held a new competition (Specia et al., 2020) on absolute evaluations of translation quality. In task 1, sentences are annotated with direct assessment (DA) scores as in the metrics task (Bojar et al., 2017).

We have been working on the metrics task with an approach that uses pre-trained sentence encoders (Shimanaka et al., 2018, 2019). Shimanaka et al. (2018) employed InferSent (Conneau et al., 2017), Quick-Thought (Logeswaran and Lee, 2018), and Universal Sentence Encoder (Cer et al., 2018) as encoders, and achieved the highest performance in all to-English language pairs of WMT18 metrics shared task (Ma et al., 2018). Subsequently, Shimanaka et al. (2019) employed BERT (Devlin et al., 2019) as an encoder to further improve the correlation with manual evaluations. In this study, we apply similar approaches to the QE task. However, to support both source and target languages, we employ XLM-RoBERTa[2] (Conneau et al., 2020), a pre-trained multilingual sentence encoder.

2 WMT20 QE Shared Task 1

In the WMT20 QE task 1 (Sentence-Level Direct Assessment), participants predict translation quality at the sentence level from pairs of source and MT output sentences. This task provides datasets for seven language pairs and sets up a multilingual track for a language-independent approach.

2.1 Datasets

Source sentences have been collected from Wikipedia for six language pairs: English–German (En-De), English–Chinese (Eh-Zh), Romanian–English (Ro-En), Estonian–English (Et-En), Nepalese–English (Ne-En), and Sinhala–English(Si-En). In addition, a combination of 75% Reddit data and 25% Wikipedia data for the Russian–English (Ru-En) language pair is provided. Organizers trained state-of-the-art neural MT models on each dataset using the fairseq toolkit (Ott et al., 2019) and generated MT output sentences.

[1] **T**okyo **M**etropolitan **U**niversity and **O**saka **U**niversity

[2] https://github.com/facebookresearch/XLM

Proceedings of the 5th Conference on Machine Translation (WMT), pages 1037–1041

Source	MT output	QE score
Its ferocious winds defoliated nearly all vegetation, splintering or uprooting thousands of trees and decimating the island's lush rainforests.	Seine wilden Winde entblätterten fast die gesamte Vegetation, zersplitterten oder entwurzelten Tausende von Bäumen und dezimierten die üppigen Regenwälder der Insel.	1.267
The Cubs tied it in the third on a triple by Ben Zobrist to knock in Daniel Murphy.	Die Cubs band es in der dritten auf einem Triple von Ben Zobrist in Daniel Murphy klopfen.	−3.760

Table 1: Examples of English-German dataset.

Three or more professional translators annotated DA scores in the range of 0-100 points for each pair of source and MT output sentences. These annotations are following the FLORES setup (Guzmán et al., 2019). The dataset consists of pairs of source and MT output sentences, z-standardized DA scores, and MT model score (log probabilities for words). Table 1 shows examples of the dataset. For each language pair, 7,000 training sets, 1,000 development sets, and 1,000 test sets are provided.

2.2 Baseline and Evaluation

The baseline system is a Predictor-Estimator model (Kim et al., 2017) implemented in OpenKiwi[3] (Kepler et al., 2019). The predictor is trained on a parallel corpus used to train the MT model, and predicts each target token from source and target contexts. And the estimator predicts the QE score from features produced by the predictor.

Participants are evaluated by Pearson's correlation metric (Pearson), mean absolute error (MAE), and root mean squared error (RMSE). A z-standardized DA score is used as a gold label.

3 TMUOU System

Our system is an ensemble model of four regression models based on XLM-RoBERTa (Conneau et al., 2020) with language tags. We first explain each base model in Section 3.1, and then introduce the ensemble model in Section 3.2. Finally, Section 3.3 describes the implementation details.

3.1 Base Models

Recently, the fine-tuning approach for masked language models (Devlin et al., 2019) has achieved the highest performance for many language understanding tasks (Wang et al., 2019). The BERT-based regression model (Shimanaka et al., 2019) also achieves high performance in the WMT metric task that estimates the DA score of translation quality (Bojar et al., 2017). We employ XLM-RoBERTa (Conneau et al., 2020), a multilingual masked language model, for this task to estimate the DA score of translation quality from pairs of source and MT output sentences.

E0 Model In this model, we fine-tune the XLM-RoBERTa in the normal way. We input sentence pairs into the model in the following format and use the special token `<s>` at the beginning of the first sentence to estimate the QE score: `<s> source </s> <s> MT output </s>`.

E0+LangTag Model To make it clear to the XLM-RoBERTa which language each sentence is in, we add a special token (LangTag) for language identification, such as `<en>`, at the beginning of each sentence. We have expanded the tokenizer and vocabulary and added the following eight LangTags: `<en> <et> <de> <ne> <ro> <ru> <si> <zh>`. An example of input to the model is as follows: `<s> <en> source </s> <s> <de> MT output </s>`.

E0+AVG Model Averaged token vector is as fruitful as the `<s>` vector at the beginning of the first sentence (Reimers and Gurevych, 2019). We concatenate the averaged token vector and the `<s>` vector to get richer information from sentence pairs.

E0+AVG+LangTag Model This model is a combination of the above models. As shown in Figure 1, we add LangTag at the beginning of each sentence and concatenate the `<s>` vector with the averaged token vector to estimate the QE score.

3.2 Ensemble Model

We ensemble four models described above to make prediction stable. A Gradient Boosting Tree (Fried-

[3]https://github.com/Unbabel/OpenKiwi

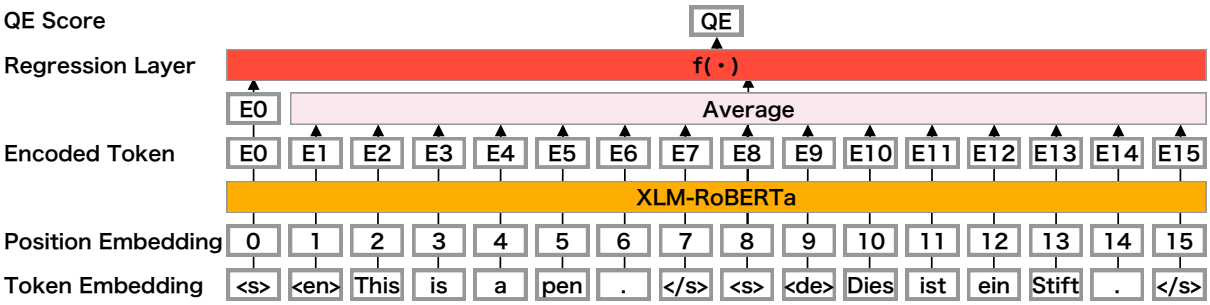


Figure 1: Overview of the TMUOU system.

	En-De	En-Zh	Ro-En	Et-En	Ne-En	Si-En	Ru-En	Multilingual
E0	0.455	**0.490**	0.860	0.747	0.742	0.646	0.693	0.662
E0+LangTag	0.419	0.465	0.874	0.744	0.763	0.648	**0.701**	0.652
E0+AVG	**0.461**	0.440	0.873	0.738	0.751	**0.658**	0.689	0.659
E0+AVG+LangTag	0.410	0.465	**0.885**	**0.764**	**0.769**	0.646	0.699	**0.663**
Ensemble	0.485	0.506	0.897	0.783	0.801	0.691	0.726	0.698

Table 2: Pearson's correlation on the development sets.

man, 2001) is trained using k-fold cross-validation on the development set with the QE scores estimated by each base model as the features. In addition to the QE scores estimated by each base model, the features of the ensemble model also include the sum of MT model scores for each output word and a one-hot vector representing the language pair.

3.3 Implementation Details

We implemented all models based on the Hugging Face (Wolf et al., 2019) XLM-RoBERTa-large model.[4] The hyper parameters are as follows: batch size is 16, weight decay is 0.01, gradient clipping norm is 5.0, dropout for the attention layers and regression layer are 0.1, max epoch is 100. We use early stopping by Pearson metric on the dev sets with patience 5. We use Adam optimizer (Kingma and Ba, 2015) with warm up. The learning rate for the optimizer is $2e^{-5}$, and we gradually decrease the learning rate by a linear scheduler.

For the ensemble model, we trained gradient boosting regressor with least square loss implemented in scikit-learn (Pedregosa et al., 2011) with 10 folds cross-validation. The hyper parameters are as follows: the initial learning rate is 0.1, the number of estimators are 100, the subsample ratio is 1.0, the criterion is mean squared error with improvement score by Friedman, the minimum amount of sample split is 2, max depth of the tree is 3.

[4]https://huggingface.co/xlm-roberta-large

	MAE	RMSE	Pearson
Bergamot-LATTE	0.408	0.527	0.718
TMUOU	**0.418**	**0.543**	**0.686**
IST and Unbabel	0.433	0.569	0.673
TransQuest	0.480	0.596	0.722
NiuTrans	0.529	0.653	0.732
WL Research	0.538	0.683	0.546
IST and Unbabel	0.547	0.719	0.583
Baseline	0.788	0.999	0.376
Bergamot-LATTE	0.895	1.062	0.489
nc	0.918	1.141	0.462

Table 3: Official results in ascending order of MAE.

4 Results

Table 2 shows the Pearson's correlation of each model on the development sets. Although there is no significant difference in the performance of the base models, the E0+AVG+LangTag model achieves higher performance in the majority of language pairs. The ensemble model achieves the highest performance in all language pairs. QE performance of to-English language pairs tends to be higher than that of from-English language pairs.

Table 3 presents the official results for a multilingual track. Participants are listed in ascending order of MAE. We submitted the ensemble model and ranked 4th in Pearson and 2nd in MAE and RMSE on a multilingual track.

5 Conclusions

We describe the TMUOU submission for the WMT20 Shared Task on Quality Estimation. Our system is an ensemble model based on XLM-RoBERTa, which takes into account averaged token vectors and language identifiers to improve performance. In the official evaluation, we ranked 4th in Pearson and 2nd in MAE and RMSE on a multilingual track.

Acknowledgments

This work was supported by JST ACT-X Grant Number JPMJAX1907, and JSPS KAKENHI Grant Number JP20K19861.

References

John Blatz, Erin Fitzgerald, George Foster, Simona Gandrabur, Cyril Goutte, Alex Kulesza, Alberto Sanchis, and Nicola Ueffing. 2004. Confidence Estimation for Machine Translation. In *Proceedings of the 20th International Conference on Computational Linguistics*, pages 315–321.

Ondřej Bojar, Yvette Graham, and Amir Kamran. 2017. Results of the WMT17 Metrics Shared Task. In *Proceedings of the Second Conference on Machine Translation*, pages 489–513.

Daniel Cer, Yinfei Yang, Sheng yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. 2018. Universal Sentence Encoder for English. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 169–174.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised Cross-lingual Representation Learning at Scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised Learning of Universal Sentence Representations from Natural Language Inference Data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186.

Erick Fonseca, Lisa Yankovskaya, André F. T. Martins, Mark Fishel, and Christian Federmann. 2019. Findings of the WMT 2019 Shared Tasks on Quality Estimation. In *Proceedings of the Fourth Conference on Machine Translation*, pages 1–12.

Jerome H. Friedman. 2001. Greedy Function Approximation: A Gradient Boosting Machine. *Annals of Statistics*, 29(5):1189–1232.

Simona Gandrabur and George Foster. 2003. Confidence Estimation for Translation Prediction. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 95–102.

Francisco Guzmán, Peng-Jen Chen, Myle Ott, Juan Pino, Guillaume Lample, Philipp Koehn, Vishrav Chaudhary, and Marc'Aurelio Ranzato. 2019. The FLORES Evaluation Datasets for Low-Resource Machine Translation: Nepali–English and Sinhala–English. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 6098–6111.

Fabio Kepler, Jonay Trénous, Marcos Treviso, Miguel Vera, and André F. T. Martins. 2019. OpenKiwi: An Open Source Framework for Quality Estimation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 117–122.

Hyun Kim, Jong-Hyeok Lee, and Seung-Hoon Na. 2017. Predictor-Estimator using Multilevel Task Learning with Stack Propagation for Neural Quality Estimation. In *Proceedings of the Second Conference on Machine Translation*, pages 562–568.

Diederik P. Kingma and Jimmy Lei Ba. 2015. Adam: A Method for Stochastic Optimization. In *Proceedings of the 3rd International Conference on Learning Representations*.

Lajanugen Logeswaran and Honglak Lee. 2018. An Efficient Framework for Learning Sentence Representations. In *Proceedings of the 6th International Conference on Learning Representations*.

Qingsong Ma, Ondřej Bojar, and Yvette Graham. 2018. Results of the WMT18 Metrics Shared Task: Both Characters and Embeddings Achieve Good Performance. In *Proceedings of the Third Conference on Machine Translation*, pages 682–701.

Qingsong Ma, Johnny Wei, Ondřej Bojar, and Yvette Graham. 2019. Results of the WMT19 Metrics Shared Task: Segment-Level and Strong MT Systems Pose Big Challenges. In *Proceedings of the Fourth Conference on Machine Translation*, pages 62–90.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A Fast, Extensible Toolkit for Sequence Modeling. In *Proceedings of*

the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 48–53.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12(85):2825–2830.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 3982–3992.

Hiroki Shimanaka, Tomoyuki Kajiwara, and Mamoru Komachi. 2018. RUSE: Regressor Using Sentence Embeddings for Automatic Machine Translation Evaluation. In *Proceedings of the Third Conference on Machine Translation*, pages 764–771.

Hiroki Shimanaka, Tomoyuki Kajiwara, and Mamoru Komachi. 2019. Machine Translation Evaluation with BERT Regressor. *arXiv:1907.12679*.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A Study of Translation Edit Rate with Targeted Human Annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas*, pages 223–231.

Lucia Specia, Frédéric Blain, Marina Fomicheva, Erick Fonseca, Vishrav Chaudhary, Francisco Guzmán, and André FT Martins. 2020. Findings of the WMT 2020 Shared Task on Quality Estimation. In *Proceedings of the Fifth Conference on Machine Translation: Shared Task Papers*.

Lucia Specia, Carolina Scarton, and Gustavo Henrique Paetzold. 2018. Quality Estimation for Machine Translation. *Synthesis Lectures on Human Language Technologies*, 11(1):1–162.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In *Proceedings of the 7th International Conference on Learning Representations*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. HuggingFace's Transformers: State-of-the-art Natural Language Processing. *arXiv:1910.03771*.

NICT Kyoto Submission for the WMT'20 Quality Estimation Task: Intermediate Training for Domain and Task Adaptation

Raphael Rubino
National Institute of Information and Communications Technology
3-5 Hikaridai, Seika-cho, Soraku-gun, Kyoto, 619-0289, Japan
raphael.rubino@nict.go.jp

Abstract

This paper describes the NICT Kyoto submission for the WMT'20 Quality Estimation (QE) shared task. We participated in *Task 2: Word and Sentence-level Post-editing Effort*, which involved Wikipedia data and two translation directions, namely English-to-German and English-to-Chinese. Our approach is based on multi-task fine-tuned cross-lingual language models (XLM), initially pre-trained and further domain-adapted through intermediate training using the translation language model (TLM) approach complemented with a novel self-supervised learning task which aim is to model errors inherent to machine translation outputs. Results obtained on both word and sentence-level QE show that the proposed intermediate training method is complementary to language model domain adaptation and outperforms the fine-tuning only approach.

1 Introduction

This paper presents the NICT Kyoto submission for the ninth edition of the quality estimation (QE) shared task organized at the fifth conference for machine translation (WMT'20). The goal of QE is to estimate the quality of machine translation (MT) output without using a translation reference. The system developed for the task and described in this paper is based on pre-trained cross-lingual language models (XLM) (Conneau and Lample, 2019), domain and task-adapted through intermediate training (Phang et al., 2018) and fine-tuned in a multi-task fashion for the sentence and word-level QE objectives.

It was shown during the QE shared task at WMT'19 (Fonseca et al., 2019) that pre-trained language models (LM) fine-tuned for QE reach state-of-the-art results at the levels of sentence and word following the predictor–estimator architecture (Kim et al., 2017) or using a fully end-to-end approach (Kepler et al., 2019; Kim et al., 2019; Zhou et al., 2019). However, fine-tuning pre-trained LMs is highly unstable when the dataset used for fine-tuning is small (Devlin et al., 2019; Zhang et al., 2020), which is usually the case for QE, as annotated datasets are scarce and expensive to produce, and WMT QE datasets are no exceptions (the shared task datasets are presented in Table 3). This fine-tuning instability might be due to neural network (NN) optimization difficulties or lack of generalization. (Mosbach et al., 2020)

To reduce fine-tuning instability of pre-trained LMs, Phang et al. (2018) introduced intermediate training, using large scale labeled data relevant to the target task in order to provide the pre-trained model with a transition step towards the final task. This approach is nonetheless limited by its reliance on annotated data for supervised learning. In our work, we propose a novel self-supervised intermediate training approach to adapt a pre-trained model to QE which does not rely on labelled data. We modify the popular masked LM objective to model simultaneously deletions and insertions in translations, two error types commonly observed in MT outputs.

Our approach is complementary to LM domain adaptation and we propose to conduct both tasks, i.e. domain and final task adaptation, jointly during intermediate training and prior to fine-tuning. More details about the intermediate training approach, including masked LM modifications and the datasets used, are presented in Section 2, followed by the QE task fine-tuning and evaluation in Section 3. Finally, a conclusion and future work are given in Section 4.

2 Intermediate Training

We describe in this Section the intermediate training process applied to the pre-trained LM used in our QE submission. This method could be applied

Proceedings of the 5th Conference on Machine Translation (WMT), pages 1042–1048

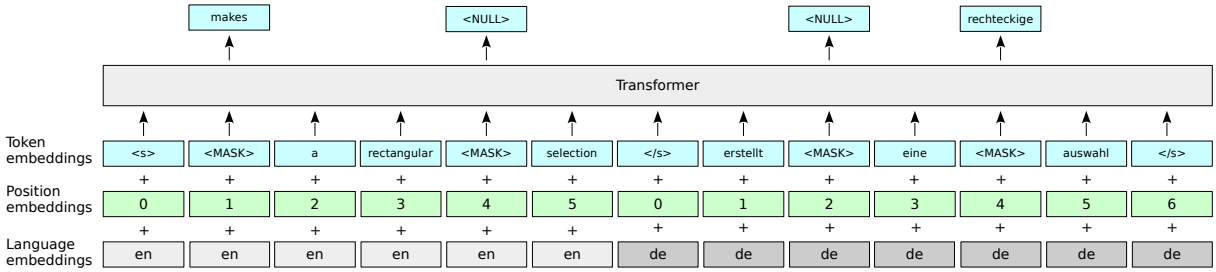


Figure 1: Intermediate self-supervised learning task based on the translation language model training objective of XLM with the addition of *NULL* tokens associated with randomly inserted *MASK* tokens.

to any pre-trained LM, but also when training a masked LM from scratch.

2.1 Approach Description

The fine-tuning of pre-trained LM has been applied to and has improved the performances of many natural language processing tasks such as grammatical sentence classification, paraphrases detection or textual entailment to name a few popular tasks. (Wang et al., 2018)

Some of the prevailing fine-tuned pretrained models studied in the literature are BERT (Devlin et al., 2019) and XLM (Conneau and Lample, 2019), among others. At the core of these approaches are similar LM techniques, using the sequentiality of languages to learn probabilities over sequences (X) of words (x_i, $i \in [0; n]$) as in $p(X) = \prod_{i=1}^{n} p(x_n|x_1, ..., x_n)$ (causal LM) or randomly masking some input tokens and learning to retrieve them based on both left and right contexts (masked LM). The masked LM approach introduced in BERT was extended in XLM to learn relations between translated sentences based on bilingual parallel corpora, integrating a new training objective called translation LM (TLM).

The TLM is particularly suited for QE, as it allows the model to learn bilingual context information when predicting masked tokens. However, fine-tuning pre-trained models was shown to be unstable with small datasets (Devlin et al., 2019), the reasons of this instability being studied recent work (Zhang et al., 2020; Mosbach et al., 2020). A proposed approach to reduce instability is to use a second stage pre-training step, between the initial LM training and the final task-oriented fine-tuning. It is based on a large amount of labeled data for a task related to the target objective. In addition to providing a *smooth* transition between initial pre-training and fine-tuning by coercing the model towards the final training objective, the intermediate step allows for domain adaptation when there is a domain mismatch between the datasets used for each training step. (Phang et al., 2018)

As a variant to the intermediate training approach, which originally makes use of labeled data, we propose a self-supervised intermediate step, alleviating the need for annotated data. We aim at combining both the domain adaptation advantage of continued training by using a dataset relevant to the final task, and target objective adaptation by modifying the masked LM approach used in the TLM model. More precisely, in addition to predicting the vocabulary masked in the input parallel sequences, we introduce *fake* masks for which a *null* token has to be predicted. This method forces the model to distinguish between missing words, which often occure in translated sentences when source words are not translated, and wrongly introduced words, similar to mistranslations when source words are wrongly translated. The proposed intermediate self-supervised learning task is illustrated in Figure 1.

2.2 Datasets and Tools

The domain and task adapted LMs used for our QE submissions are based on the pre-trained XLM model made available as a checkpoint in the HuggingFace Transformers library (Wolf et al., 2019), including 15 languages and trained using masked TLM.[1]. This model uses a sub-word vocabulary of 95k tokens shared between all languages, $1,024$ dimensions embeddings, learned language and position embeddings, 12 transformer blocks including 16 heads self-attention layers and $4,096$ dimensions feed-forward layers with Gaussian Error Linear Units (GELU) activation functions. The model has a total of approx. 249M parameters. The train-

[1]Model called *xlm-mlm-tlm-xnli15-1024* and available at https://github.com/huggingface/transformers

	Sentence	Source Token	Source Type	MT Token	MT Type
		EN–DE			
Train	7.8M	129.6M	2.2M	124.5M	4.0M
Valid.	8.0k	112.7k	34.9k	115.0k	37.2k
		EN–ZH			
Train	3.3M	61.0M	0.3M	102.0M	8.0k
Valid.	8.0k	113.0k	34.8k	0.3M	3.8k

Table 1: Number of sentences, source tokens, source types, MT tokens and MT types in the training and validation sets used for LM intermediate training. Tokens and types denote words for English and German, and characters for Chinese, including numbers and punctuation marks.

ing objective is similar to the original TLM, except for an additional token in the vocabulary corresponding to the *null* token. We ran intermediate training for English–German and English–Chinese language pairs separately. The code used to conduct intermediate training was developed in-house on top of the HuggingFace Transformers library and written in PyTorch (Adam et al., 2017).

The datasets used for intermediate training are detailed in Table 1. We relied on the parallel data provided by the QE shared task organizers for English–German and English– Chinese, after selecting the most relevant sentence pairs based on their coverage of the source and MT output vocabulary extracted from the QE training, validation and test data. Using the test source and corresponding MT output is a limitation of the models presented in this paper, as a commercial QE system based on this method would require re-training when QE scores have to be produced for unseen data. However, our data filtering approach is still reliable without using the test set, as it is shown in Rubino and Sumita (2020). To remove noisy parallel sentences from the data used for intermediate training, we only kept sentence pairs containing a minimum of 3 tokens in the source and target sentences and with at least 40% of their tokens longer than 4 characters being in the QE vocabulary. In addition for the English–German LMs, we used the WikiMatrix corpus (Schwenk et al., 2019) made available by the WMT organizers for the news translation task.[2]

[2] http://data.statmt.org/wmt20/translation-task/WikiMatrix/

2.3 Training Procedure

Hyper-parameters specific to masked LMs, such as the amount of masked tokens per sequence, or more general to NNs, such as the optimizer learning-rate, have to be set prior to training. While the latter ones were suggested in previous work (Devlin et al., 2019; Conneau and Lample, 2019), we define and propose some values for the former ones in this paper. We trained a total of 8 masked LMs with variations in hyper-parameters, keeping checkpoints for each model based on the loss obtained on the validation set and at the end of every epoch. General and masked LM specific hyper-parameters are described in the following subsections and a summary of the trained masked LMs is presented in Table 2.

Note that we followed a token sampling similar to the one in XLM (Conneau and Lample, 2019), i.e., a first hyper-parameter is dedicated to the percentage of tokens to randomly select from a text sequence (noted *sample* in Table 2), a second hyper-parameter (noted *mask*) is allocated to the percentage of initially selected tokens which are replaced by a special *mask* token, a third hyper-parameter (noted *rand.*) is assigned to the percentage of initially selected tokens which are not replaced by *mask* but by tokens randomly sampled from the vocabulary. Finally, we introduce a fourth hyper-parameter, dedicated to the percentage of additional *mask* tokens introduced in a text sequence and corresponding to the *null* token.

2.3.1 General Hyper-parameters

All our masked LMs trained for the QE task used the *AdamW* optimizer (Loshchilov and Hutter, 2017) with the following parameters: $\beta_1 = 0.9$, $\beta_2 = 0.98$, $\epsilon = 1e^{-8}$ and weight decay set at $1e^{-8}$. The learning rate followed a linear schedule with a warm-up period during the first 4k steps to reach a maximum value of $5e^{-5}$ or $1e^{-4}$ depending on the model (as detailed in Table 2), then decayed until the model reached 100k steps. Depending on the model, the batch size was set to 32 or 64 with gradient accumulation set to 16 batches, respectively simulating batch sizes of 512 and $1,024$ pairs of source and target sequences.

2.3.2 Masked LM Hyper-parameters

We experimented with various percentages of tokens in pairs of text sequences to randomly sample initially, from 10% to 20%. From this selection rate, we made variations in how many were

id	sample	mask	rand.	fake	src	tgt	bsz	lr
1	15	80	50	0			32	$1e^{-4}$
2	15	25	95	0			32	$1e^{-4}$
3	10	50	100	10	✓	✓	64	$5e^{-5}$
4	15	10	90	20	✓	✓	32	$1e^{-4}$
5	15	80	50	25	✓	✓	64	$5e^{-5}$
6	15	50	100	25		✓	64	$5e^{-5}$
7	15	50	100	25	✓	✓	64	$5e^{-5}$
8	20	50	100	20	✓	✓	64	$5e^{-5}$

Table 2: Masked LMs with different hyper-parameters chosen for the intermediate training step. The model identifier is denoted in the column *id*, *sample* indicates the percentage of tokens randomly sampled from the sentence pairs, *mask* denotes the percentage of sampled tokens replaced by the *mask* token, *rand.* corresponds to the percentage of tokens replaced by a randomly sampled token from the vocabulary, *fake* is the percentage of masked token corresponding to the *null* token, *src* and *trg* indicate if fake masks are introduced in the source or target sentences respectively, *bsz* is the batch size and *lr* is the learning rate.

		Source		MT	
	Sentence	Token	Type	Token	Type
		EN–DE			
Train	7.0k	115.0k	25.4k	112.3k	28.1k
Valid.	1.0k	16.5k	6.4k	16.2k	6.7k
Test	1.0k	16.4k	6.4k	16.1k	6.5k
		EN–ZH			
Train	7.0k	115.6k	25.1k	214.6k	3.1k
Valid.	1.0k	16.3k	6.3k	30.5k	2.2k
Test	1.0k	16.8k	6.4k	30.1k	2.3k

Table 3: Number of sentences, source tokens, source types, MT tokens and MT types in the training, validation and test sets for the WMT'20 QE Task 2: Word and Sentence-level Post-editing Effort. Tokens and types denote words for English and German, and characters for Chinese, including numbers and punctuation marks.

replaced by the token *mask*: from 10% to 80%. From the remaining tokens initially sampled and not masked, from 50% to 100% of them were replaced by another token sampled randomly from the vocabulary. Finally, the remaining tokens initially sampled but not masked nor replaced were left unchanged. The percentage of fake masks (corresponding to the *null* token) was varied from 0% to 25%, additionally to the percentage of tokens randomly sampled intially during the first step. We also investigated the introduction of fake masks in the source or target sequences only, and in both source and target sequences.

3 QE Fine-tuning

The objective of fine-tuning masked LMs for QE is to predict sentence-level human translation edit rate (HTER) and word-level *good* and *bad* classes.[3] Note that in our models, for the target sequence word-level QE, we considered gaps between target words (missing translations) as part of the target sequence and did not use a loss nor a training objective specific to gaps.

3.1 Dataset

We used the training, validation and test sets released by the shared task organizers without any additional annotated data. Details about the official QE dataset are presented in Table 3.

3.2 Training Procedure

Our models presented in this paper were inspired by the approach of (Kim et al., 2019), however, we use XLM instead of BERT. We added two parallel outputs on top of XLM composed of parametrised linear layers. The first output layer corresponds to the word-level QE task, takes as input the word-level final hidden states given by XLM, and outputs word-level probabilities for the two classes (*OK* and *BAD*) using a softmax function. The second output layer corresponds to the sentence-level QE task, takes as input the final hidden state of the first token in a sentence pair (noted $<s>$ in Figure 1) given by XLM, and outputs a sequence-level probability using a sigmoid function. To compute the multi-task loss function, we first computed two loss functions separately, namely cross-entropy and mean squared error for the word-level and sentence-level QE respectively, based on the network predictions and the training gold labels. The two losses were then summed without weights to compose the final loss.

We chose the masked LMs to fine-tune based on the validation (presented in Table 1) loss and at the end of epoch 5 for English–German and epoch 10 for English–Chinese (the latter models were faster to train due to the smaller LM intermediate training data size). Thus, 2 checkpoints were kept for each of the 8 models presented in Table 2. In order to find good hyper-parameters to fine-tune the masked LMs for QE and because the QE datasets are relatively small, we conducted a grid-search among

[3] More details about the WMT'20 QE Task 2: Word and Sentence-level Post-editing Effort are available at http://www.statmt.org/wmt20/quality-estimation-task.html

id	*EN–DE* $r \uparrow$	MAE↓	RMSE↓	*EN–ZH* $r \uparrow$	MAE↓	RMSE↓
0	0.221	0.159	0.198	0.461	0.155	0.193
0*	0.564	0.167	0.214	0.604	0.151	0.193
1	0.566	0.173	0.224	0.664	0.135	0.167
2	0.571	**0.138**	0.177	0.658	0.128	0.162
3	0.593	0.161	0.208	0.668	0.145	0.178
4	0.578	0.173	0.224	0.638	0.135	0.170
5	0.598	0.151	0.197	0.663	0.130	0.164
6	**0.605**	0.167	0.218	**0.669**	**0.125**	**0.158**
7	0.594	0.146	0.190	0.665	0.126	**0.158**
8	0.601	**0.138**	**0.176**	0.657	0.144	0.178

Table 4: Sentence-level predicted post-editing effort on the official WMT'20 QE validation set. The *id* column refers to the Model ID as presented in Table 2. The *id* 0 denotes the out-of-the-box XLM checkpoint without domain or task adaptation through intermediate training and without QE fine-tuning. The *id* 0* denotes the QE fine-tuned XLM checkpoint without domain or task adaptation through intermediate training.

the following hyper-parameters: masked LM and output layer learning rates, dropout rate, using or not class weights for the softmax function, and finally the decay rate applied to the discriminative fine-tuning approach (Howard and Ruder, 2018). During hyper-parameter search and training of the final models, the batch size was set to 64 sequence pairs and the learning rate was warmed-up linearly for 200 steps. The remaining hyper-parameters were set to values identical to the ones presented in Section 2.3.

3.3 Evaluation

We present in this Section the results obtained during our experiments, first on the official validation set and then on the official test set, based on the masked LMs presented in Table 2. For the sentence-level post-editing effort prediction, the official primary metric was the Pearson correlation coefficient (r) and two supplementary metrics were used: mean absolute error (MAE) and root mean squared error (RMSE). For the word-level binary classes prediction, the official primary metric was the Matthews correlation coefficient (MCC) and supplementary F-measures for the *OK* class and for the *BAD* class were used. The word-level evaluation was conducted on source and target sequences separately. The results obtained on the sentence-level task are presented in Table 4 and the results obtained on the word-leve task are presented in Table 5. For the latter, we present a single F-score for both *OK* and *BAD* classes by multiplying individual F-scores (similarily to the *F1 mult* score used during the WMT'19 QE task (Fonseca et al., 2019)).

Results obtained at the sentence-level (Table 4) show that both domain adaptation and fake-masking are useful as an intermediate training task prior to QE fine-tuning. The best results according to Pearson's r are reached by the model #6 for the two language pairs. This model has an equal amount of masked and randomly replaced tokens, and fake masks are inserted in target sequences only. The same model reaches the best results for the EN–ZH language pair for all the metrics while there is no best performing model on all metrics for the EN–DE pair. When comparing the models obtained with configurations #1 and #5, which differ mainly on the introduction of fake masks for the latter, best performances are reached by model #5 as indicated by the three metrics, showing that fake masking is helpful in predicting sentence-level post-editing effort. However, the batch size and the learning rate also differ for these two configurations. A more consistant ablation study allowing for a fair comparison between configurations with and without fake masking is presented in Rubino and Sumita (2020).

Experiments on the word-level results (Table 5) show that introducing fake-masks is useful for the EN–DE language pair on both source and target text sequences, as the best performances according to both metrics are reached by models #5, #7 and #8. The introduction of fake masks in model #5, compared to model #1 which does not have fake masks, show that this method is helpful for this language pair at predicting word-level quality estimation. However, this is not the case on the source side for EN–ZH, where model #1 reaches the best results in terms of MCC and F1. This model does not involve fake-masking but only domain adaptation. On the target side, however, the best results according to both metrics are reached by models involving fake-masking, namly models #3, #6 and #7 with 0.566 MCC and 0.604 F1.

Our final submission to the shared task was composed of an ensemble of all the checkpoints for all the models, i.e. 32 models per language pair and QE task (8 pre-trained models, fine-tuning checkpoints based on validation loss, primary metric score and best epoch). We present in Table 6 and Table 7 the official results obtained by our final submission ensembles on the test set as reported by the shared task organizers on the sentence-level

	EN–DE				*EN–ZH*			
	Source		Target		Source		Target	
id	MCC↑	F1	MCC↑	F1	MCC↑	F1	MCC↑	F1
0	0.207	0.314	0.351	0.387	0.192	0.344	0.511	0.536
0⋆	0.326	0.407	0.432	0.461	0.324	0.436	0.564	0.600
1	0.306	0.398	0.438	0.480	**0.347**	**0.452**	0.560	0.598
2	0.312	0.397	0.434	0.476	0.338	0.448	0.558	0.598
3	0.329	0.417	0.438	0.478	0.322	0.435	**0.566**	**0.604**
4	0.309	0.395	0.440	0.482	0.313	0.402	0.564	0.600
5	**0.347**	0.413	**0.451**	0.487	0.322	0.437	0.563	0.602
6	0.330	0.415	0.442	0.482	0.338	0.444	**0.566**	0.603
7	0.331	0.403	**0.451**	**0.490**	0.328	0.441	0.565	**0.604**
8	0.342	**0.425**	0.449	0.489	0.310	0.424	0.553	0.592

Table 5: Word-level predicted binary classes on the official WMT'20 QE validation set. The *id* column refers to the Model ID as presented in Table 2. The *id* 0 denotes the out-of-the-box XLM checkpoint without domain or task adaptation through intermediate training and without QE fine-tuning. The *id* 0⋆ denotes the QE fine-tuned XLM checkpoint without domain or task adaptation through intermediate training.

rank	Pearson's r ↑	MAE ↓	RMSE ↓
	EN–DE		
5	0.615	0.151	0.197
	EN–ZH		
3	0.643	0.129	0.161

Table 6: Official sentence-level WMT'20 QE Task 2 results on the test set as reported by the shared task organizer. The column *rank* indicates the ranking of our submission among other participants according to the primary metric (Pearson's r).

rank	MCC↑	$F1_{BAD}$ ↑	$F1_{OK}$ ↑
	Source EN–DE		
3	0.353	0.537	0.806
	Target EN–DE		
3	0.485	0.568	0.916
	Source EN–ZH		
1	0.336	0.668	0.669
	Target EN–ZH		
2	0.582	0.704	0.878

Table 7: Official word-level WMT'20 QE Task 2 results on the test set as reported by the shared task organizer. The column *rank* indicates the ranking of our submission among other participants according to the primary metric (MCC).

and word-level tasks respectively.

4 Conclusion

We have presented in this paper the NICT Kyoto submission for the WMT'20 QE shared task on predicting post-editing effort at the sentence and word-level. Our submissions consisted of ensembles of several fine-tuned masked LMs, pre-trained using the translation LM objective, domain and task adapted in a self-supervised fashion using domain-relevant data and a modified masking approach during intermediate training.

This novel intermediate training objective allows for a *smooth* transition from a pre-trained masked LM towards the final QE task without requiring annotated data. We have shown empirically that both domain and task adaptation reach good results compared to out-of-the-box pre-trained models and compared to fine-tuning only. Our final submissions were ranked among the top systems both at the sentence and word-level for two language pairs.

Acknowledgment

A part of this work was conducted under the commissioned research program "Research and Development of Advanced Multilingual Translation Technology" in the "R&D Project for Information and Communications Technology (JPMI00316)" of the Ministry of Internal Affairs and Communications (MIC), Japan. We would like to thank the reviewers for their insightful comments and suggestions.

References

Paszke Adam, Gross Sam, Chintala Soumith, Chanan Gregory, Yang Edward, D Zachary, Lin Zeming, Desmaison Alban, Antiga Luca, and Lerer Adam. 2017. Automatic Differentiation in PyTorch. In *Proceedings of NIPS Autodiff Workshop*.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual Language Model Pretraining. In *Proceedings of NeurIPS*, pages 7057–7067.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of NAACL*, pages 4171–4186.

Erick Fonseca, Lisa Yankovskaya, André F. T. Martins, Mark Fishel, and Christian Federmann. 2019. Findings of the WMT 2019 Shared Tasks on Quality Estimation. In *Proceedings of WMT*, pages 1–12.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*.

Fabio Kepler, Jonay Trénous, Marcos Treviso, Miguel Vera, António Góis, M. Amin Farajian, António V. Lopes, and André F. T. Martins. 2019. Unbabel's Participation in the WMT19 Translation Quality Estimation Shared Task. In *Proceedings of WMT*, pages 80–86.

Hyun Kim, Jong-Hyeok Lee, and Seung-Hoon Na. 2017. Predictor-estimator Using Multilevel Task Learning with Stack Propagation for Neural Quality Estimation. In *Proceedings of WMT*, pages 562–568.

Hyun Kim, Joon-Ho Lim, Hyun-Ki Kim, and Seung-Hoon Na. 2019. QE BERT: Bilingual BERT Using Multi-task Learning for Neural Quality Estimation. In *Proceedings of WMT*, pages 87–91.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Marius Mosbach, Maksym Andriushchenko, and Dietrich Klakow. 2020. On the stability of fine-tuning bert: Misconceptions, explanations, and strong baselines. *arXiv preprint arXiv:2006.04884*.

Jason Phang, Thibault Févry, and Samuel R Bowman. 2018. Sentence encoders on stilts: Supplementary training on intermediate labeled-data tasks. *arXiv preprint arXiv:1811.01088*.

Raphael Rubino and Eiichiro Sumita. 2020. Intermediate Self-supervised Learning for Machine Translation Quality Estimation. In *Proceedings of the 28th International Conference on Computational Linguistics*.

Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán. 2019. Wikimatrix: Mining 135m parallel sentences in 1620 language pairs from wikipedia. *arXiv preprint arXiv:1907.05791*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. HuggingFace's Transformers: State-of-the-art Natural Language Processing. *ArXiv*, abs/1910.03771.

Tianyi Zhang, Felix Wu, Arzoo Katiyar, Kilian Q Weinberger, and Yoav Artzi. 2020. Revisiting few-sample bert fine-tuning. *arXiv preprint arXiv:2006.05987*.

Junpei Zhou, Zhisong Zhang, and Zecong Hu. 2019. SOURCE: SOURce-Conditional Elmo-style Model for Machine Translation Quality Estimation. In *Proceedings of WMT*, pages 108–113.

TransQuest at WMT2020: Sentence-Level Direct Assessment

Tharindu Ranasinghe[♢], **Constantin Orăsan**[♡] **and Ruslan Mitkov**[♢]
[♢]Research Group in Computational Linguistics, University of Wolverhampton, UK
[♡]Centre for Translation Studies, University of Surrey, UK
{t.d.ranasinghehettiarachchige, r.mitkov}@wlv.ac.uk
c.orasan@surrey.ac.uk

Abstract

This paper presents the team TransQuest's participation in Sentence-Level Direct Assessment shared task in WMT 2020. We introduce a simple QE framework based on cross-lingual transformers, and we use it to implement and evaluate two different neural architectures. The proposed methods achieve state-of-the-art results surpassing the results obtained by OpenKiwi, the baseline used in the shared task. We further fine tune the QE framework by performing ensemble and data augmentation. Our approach is the winning solution in all of the language pairs according to the WMT 2020 official results.

1 Introduction

The goal of quality estimation (QE) systems is to determine the quality of a translation without having access to a reference translation. This makes it very useful in translation workflows where it can be used to determine whether an automatically translated sentence is good enough to be used for a given purpose, or if it needs to be shown to a human translator for translation from scratch or postediting (Kepler et al., 2019). Quality estimation can be done at different levels: document level, sentence level and word level (Ive et al., 2018). This paper presents TransQuest, a sentence-level quality estimation framework which is the winning solution in all the language pairs in the WMT 2020 Sentence-Level Direct Assessment shared task (Specia et al., 2020).

In the past, high preforming quality estimation systems such as QuEst (Specia et al., 2013) and QuEst++ (Specia et al., 2015) were heavily dependent on linguistic processing and feature engineering. These features were fed into traditional machine-learning algorithms like support vector regression and randomised decision trees (Specia et al., 2013), which then determined the quality of a translation. Even though, these approaches provide good results, they are no longer the state of the art, being replaced in recent years by neural-based QE systems which usually rely on little or no linguistic processing. For example the best-performing system at the WMT 2017 shared task on QE was POSTECH, which is purely neural and does not rely on feature engineering at all (Kim et al., 2017).

In order to achieve high results, approaches such as POSTECH require extensive pre-training, which means they depend on large parallel data and are computationally intensive (Ive et al., 2018). TransQuest, our QE framework removes this dependency on large parallel data by using crosslingual embeddings (Ranasinghe et al., 2020) that are already fine-tuned to reflect properties between languages (Ruder et al., 2019). Ranasinghe et al. (2020) show that by using them, TransQuest eases the burden of having complex neural network architectures, which in turn entails a reduction of the computational resources. That paper also shows that TransQuest performs well in transfer learning settings where it can be trained on language pairs for which we have resources and applied successfully on less resourced language pairs.

The remainder of the paper is structured as follows. The dataset used in the competition is briefly discussed in Section 2. In Section 3 we present the TransQuest framework and the methodology employed to train it. This is followed by the evaluation results and their discussion in Section 4. The paper finishes with conclusions and ideas for future research directions.

2 Dataset

The dataset for the Sentence-Level Direct Assessment shared task is composed of data

Proceedings of the 5th Conference on Machine Translation (WMT), pages 1049–1055

extracted from Wikipedia for six language pairs, consisting of high-resource languages English-German (En-De) and English-Chinese (En-Zh), medium-resource languages Romanian-English (Ro-En) and Estonian-English (Et-En), and low-resource languages Sinhala-English (Si-En) and Nepalese-English (Ne-En), as well as a a Russian-English (Ru-En) dataset which combines articles from Wikipedia and Reddit (Specia et al., 2020). Each language pair has 7,000 sentence pairs in the training set, 1,000 sentence pairs in the development set and another 1,000 sentence pairs in the testing set. Each translation was rated with a score between 0 and 100 according to the perceived translation quality by at least three translators (Fomicheva et al., 2020). The DA scores were standardised using the z-score. The quality estimation systems have to predict the mean DA z-scores of the test sentence pairs (Specia et al., 2020).

3 Methodology

This section presents the methodology used to develop our quality estimation methods. Our methodology is based on TransQuest our recently introduced QE framework (Ranasinghe et al., 2020). We first briefly describe the neural network architectures TransQuest proposed, followed by the training details. More details about the framework can be found in (Ranasinghe et al., 2020).

3.1 Neural Network Architectures

The *TransQuest* framework that is used to implement the two architectures described here relies on the XLM-R transformer model (Conneau et al., 2020) to derive the representations of the input sentences (Ranasinghe et al., 2020). The XLM-R transformer model takes a sequence of no more than 512 tokens as input, and outputs the representation of the sequence. The first token of the sequence is always [CLS] which contains the special embedding to represent the whole sequence, followed by embeddings acquired for each word in the sequence. As shown below, proposed neural network architectures of TransQuest can utilise both the embedding for the [CLS] token and the embeddings generated for each word (Ranasinghe et al., 2020). The output of the transformer (or transformers for **SiameseTransQuest** described below), is fed into a simple output layer which is used to estimate the quality of translation. The way the XLM-R transformer is used and the output layer are different in the two instantiations of the framework. We describe each of them below. The fact that TransQuest does not rely on a complex output layer makes training its architectures much less computationally intensive than alternative solutions. The *TransQuest* framework is open-source, which means researchers can easily propose alternative architectures to the ones TransQuest presents (Ranasinghe et al., 2020).

Both neural network architectures presented below use the pre-trained XLM-R models released by HuggingFace's model repository (Wolf et al., 2019). There are two versions of the pre-trained XLM-R models named XLM-R-base and XLM-R-large. Both of these XLM-R models cover 104 languages (Conneau et al., 2020), potentially making it very useful to estimate the translation quality for a large number of language pairs.

TransQuest implements two different neural network architectures (Ranasinghe et al., 2020) to perform sentence-level translation quality estimation as described below. The architectures are presented in Figure 1.

1. **MonoTransQuest** (**MTransQuest**): The first architecture proposed uses a single XLM-R transformer model and is shown in Figure 1a. The input of this model is a concatenation of the original sentence and its translation, separated by the [SEP] token. TransQuest proposes three pooling strategies for the output of the transformer model: using the output of the [CLS] token (`CLS`-strategy); computing the mean of all output vectors of the input words (`MEAN`-strategy); and computing a max-over-time of the output vectors of the input words (`MAX`-strategy) (Ranasinghe et al., 2020). The output of the pooling strategy is used as the input of a softmax layer that predicts the quality score of the translation. TransQuest used mean-squared-error loss as the objective function (Ranasinghe et al., 2020). Similar to Ranasinghe et al. (2020), the early experiments we carried out demonstrated that the `CLS`-strategy leads to better results than the other two strategies for this architecture. Therefore, we used the embedding of the [CLS] token as the input of a softmax layer.

2. **SiameseTransQuest** (**STransQuest**): The second approach proposed in TransQuest

relies on the Siamese architecture depicted in Figure 1b which has shown promising results in monolingual semantic textual similarity tasks (Reimers and Gurevych, 2019; Ranasinghe et al., 2019). For this, we fed the original text and the translation into two separate XLM-R transformer models. Similarly to the previous architecture, we experimented with the same three pooling strategies for the outputs of the transformer models (Ranasinghe et al., 2020). TransQuest then calculates the cosine similarity between the two outputs of the pooling strategy. TransQuest used mean-squared-error loss as the objective function. Similar to Ranasinghe et al. (2020) in the initial experiments we carried out with this architecture the `MEAN`-strategy showed better results than the other two strategies. For this reason, we used the `MEAN`-strategy for our experiments. Therefore, cosine similarity is calculated between the mean of all output vectors of the input words produced by each transformer.

3.2 Training Details

We used the same set of configurations suggested in Ranasinghe et al. (2020) for all the language pairs evaluated in this paper in order to ensure consistency between all the languages. This also provides a good starting configuration for researchers who intend to use TransQuest on a new language pair. In both architectures, we used a batch-size of eight, Adam optimiser with learning rate $2e{-}5$, and a linear learning rate warm-up over 10% of the training data. The models were trained using only training data. Furthermore, they were evaluated while training using an evaluation set that had one fifth of the rows in training data. We performed early stopping if the evaluation loss did not improve over ten evaluation rounds. All of the models were trained for three epochs. For some of the experiments, we used an Nvidia Tesla K80 GPU, whilst for others we used an Nvidia Tesla T4 GPU. This was purely based on the availability of the hardware and it was not a methodological decision.

3.3 Implementation Details

The TransQuest framework was implemented using Python 3.7 and PyTorch 1.5.0. To integrate the functionalities of the transformers we used the version 3.0.0 of the HuggingFace's Transformers library. The implemented framework is available on GitHub[1].

4 Evaluation, Results and Discussion

This section presents the evaluation results of our architectures and the fine tuning strategies that can be used to improve the results. We first evaluate the TransQuest framework with the default setting (Section 4.1). Next we evaluate an ensemble setting of TransQuest in Section 4.2. We finally assess the performance of TransQuest with augmented data. We conclude the section with a discussion of the results.

The evaluation metric used was the Pearson correlation (r) between the predictions and the gold standard from the test set, which is the most commonly used evaluation metric in WMT quality estimation shared tasks (Specia et al., 2018; Fonseca et al., 2019). We report the Pearson correlation values that we obtained from CodaLab, the hosting platform of the WMT 2020 QE shared task. As a baseline we compare our results with the performance of OpenKiwi as reported by the task organisers (Specia et al., 2020).

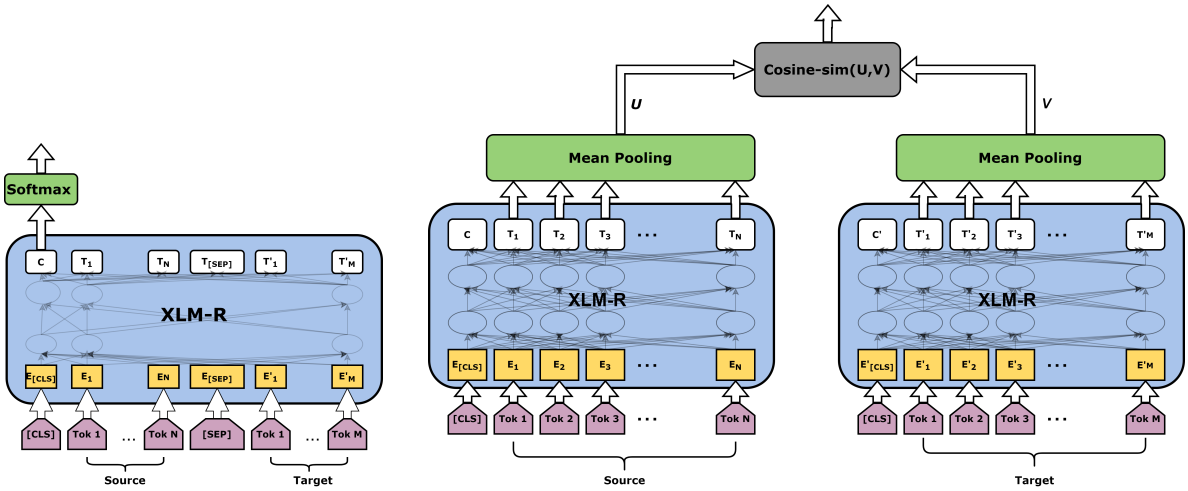
4.1 TransQuest with Default settings

The first evaluation we carried out was for the default configurations of the TransQuest framework where we used the training set of each language to build a quality estimation model using XLM-R-large transformer model and we evaluated it on a test set from the same language.

The results for each language with *default* settings are shown in row I of Table 1. The results indicate that both architectures proposed in *TransQuest* outperform the baseline, OpenKiwi, in all the language pairs. From the two architectures, *MTransQuest* performs slightly better than *STransQuest*.

As shown in Table 1, *MTransQuest* gained $\approx$ 0.2-0.3 Pearson correlation boost over OpenKiwi in all the language pairs. Additionally, *MTransQuest* achieves $\approx$ 0.4 Pearson correlation boost over OpenKiwi in the low-resource language pair Ne-En.

[1]TransQuest GitHub repository - `https://github.com/tharindudr/transQuest`



(a) *MTransQuest* architecture (b) *STransQuest* Architecture

Figure 1: Two architectures of the *TransQuest* framework.

		Low-resource		Mid-resource			High-resource	
	Method	Si-En	Ne-En	Et-En	Ro-En	Ru-En	En-De	En-Zh
I	MTransQuest	0.6525	0.7914	0.7748	0.8982	0.7734	0.4669	0.4779
	STransQuest	0.5957	0.7081	0.6804	0.8501	0.7126	0.3992	0.4067
II	MTransQuest-base	0.6412	0.7823	0.7651	0.8715	0.7593	0.4421	0.4593
	STransQuest-base	0.5773	0.6853	0.6692	0.8321	0.6962	0.3832	0.3975
III	MTransQuest ⊗	0.6661	0.8023	0.7876	0.8988	0.7854	0.4862	0.4853
	STransQuest ⊗	0.6001	0.7132	0.6901	0.8629	0.7248	0.4096	0.4159
IV	MTransQuest ⊗ - Aug	**0.6849**	**0.8222**	**0.8240**	**0.9082**	**0.8082**	**0.5539**	**0.5373**
	STransQuest ⊗ - Aug	0.6241	0.7354	0.7239	0.8621	0.7458	0.4457	0.4658
V	OpenKiwi	0.3737	0.3860	0.4770	0.6845	0.5479	0.1455	0.1902

Table 1: Pearson (r) correlation between *TransQuest* algorithm predictions and human DA judgments. Best results for each language (any method) are marked in bold. Rows I, II, III and IV indicate the different settings of *TransQuest*, explained in Sections 4.1-4.3. OpenKiwi baseline results are in Row V.

4.2 TransQuest with Ensemble

Transformers have been proven to provide better results when experimented with ensemble techniques (Xu et al., 2020). In order to improve the results of TransQuest we too followed an ensemble approach which consisted of two steps. We conducted these steps for both architectures in TransQuest.

1. We train TransQuest using the pre-trained XLM-R-base transformer model instead of the XLM-R-large transformer model in the TransQuest default setting. We report the results from the two architectures from this step in row II of Table 1 as MTransQuest-base and STransQuest-base.
2. We perform a weighted average ensemble for the output of the default setting and the output we obtained from step 1. We experimented on weights 0.8:0.2, 0.6:0.4, 0.5:0.5 on the output of the default setting and output from the step 1 respectively. Since the results we got from XLM-R-base transformer model are slightly worse than the results we got from default setting we did not consider the weight combinations that gives higher weight to XLM-R-base transformer model results. We obtained best results when we used the weights 0.8:0.2. We report the results from the two architectures from this step in row III of Table 1 as MTransQuest ⊗ and STransQuest ⊗.

As shown in Table 1 both architectures in TransQuest with ensemble setting gained ≈ 0.01-0.02 Pearson correlation boost over the default settings for all the language pairs.

4.3 TransQuest with Data Augmentation

All of the languages had 7,000 training instances that we used in the above mentioned settings in TransQuest. To experiment how TransQuest performs with more data, we trained TransQuest on a data augmented setting. Alongside the training, development and testing datasets, the shared task organisers also provided the parallel sentences which were used to train the neural machine translation system in each language. In the data augmentation setting, we added the sentence pairs from that neural machine translation system training file to training dataset we used to train TransQuest. In order to find the best setting for the data augmentation we experimented with adding 1000, 2000, 3000, up to 5000 sentence pairs randomly. Since the ensemble setting performed better than the default setting of TransQuest, we conducted this data augmentation experiment on the ensemble setting. We assumed that the sentence pairs added from the neural machine translation system training file have maximum translation quality.

Up to 2000 sentence pairs the results continued to get better. However, adding more than 2000 sentence pairs did not improve the results. We did not experiment with adding any further than 5000 sentence pairs to the training set since the timeline of the competition was tight. We were also aware that adding more sentence pairs with the maximum translation quality to the training file will make it imbalance and affect the performance of the machine learning models negatively. We report the results from the two architectures from this step in row IV of Table 1 as MTransQuest ⊗-Aug and STransQuest ⊗-Aug.

This setting provided the best results for both architectures in TransQuest for all of the language pairs. As shown in Table 1 both architectures in TransQuest with the data augmentation setting gained ≈ 0.01-0.09 Pearson correlation boost over the default settings for all the language pairs. Additionally, *MTransQuest ⊗-Aug* achieves ≈ 0.09 Pearson correlation boost over default MTransQuest in the high-resource language pair En-De.

4.4 Error analysis

In an attempt to better understand the performance and limitations of *TransQuest* we carried out an error analysis on the results obtained on Romanian - English and Sinhala - English. The choice of language pairs we analysed was determined by the availability of native speakers to perform this analysis. We focused on the cases where the difference between the predicted score and expected score was the greatest. This included both cases where the predicted score was underestimated and overestimated.

Analysis of the results does not reveal very clear patterns. The largest number of errors seem to be caused by the presence of named entities in the source sentences. In some cases these entities are mishandled during the translation. The resulting sentences are usually syntactically correct, but semantically odd. Typical examples are *RO: În urmă explorărilor Căpitanului James Cook, Australia și Noua Zeelandă au devenit ținte ale colonialismului britanic. (As a result of Captain James Cook's explorations, Australia and New Zealand have become the targets of British colonialism.) - EN: Captain James Cook, Australia and New Zealand have finally become the targets of British colonialism.* (expected: -1.2360, predicted: 0.2560) and *RO: O altă problemă importantă cu care trupele Antantei au fost obligate să se confrunte a fost malaria. (Another important problem that the Triple Entente troops had to face was malaria.)* - EN: *Another important problem that Antarctic troops had to face was malaria.* (expected: 0.2813, predicted: -0.9050). In the opinion of the authors of this paper, it is debatable whether the expected scores for these two pairs should be so different. Both of them have obvious problems and cannot be clearly understood without reading the source. For this reason, we would expect that both of them have low scores. Instances like this also occur in the training data. As a result of this, it may be that *TransQuest* learns contradictory information, which in turn leads to errors at the testing stage.

A large number of problems are caused by incomplete source sentences or input sentences with noise. For example the pair *RO: thumbright250pxDrapelul cu fâșiile în poziție verticală (The flag with strips in upright position) - EN: ghtghtness 250pxDrapel with strips in upright position* has an expected score of 0.0595, but

our method predicts -0.9786. Given that only *ghtghtness 250pxDrapel* is wrong in the translation, the predicted score is far too low. In an attempt to see how much this noise influences the result, we run the system with the pair *RO: Drapelul cu fâșiile în poziție verticală - EN: Drapel with strips in upright position*. The prediction is 0.42132, which is more in line with our expectations given that one of the words is not translated.

Similar to Ro-En, in Si-En the majority of problems seem to be caused by the presence of named entities in the source sentences. For an example in the English translation: *But the disguised Shiv will help them securely establish the statue.* (expected: 1.3618, predicted: -0.008), the correct English translation would be *But the disguised Shividru will help them securely establish the statue.*. Only the named entity *Shividru* is translated incorrectly, therefore the annotators have annotated the translation with a high quality. However TransQuest fails to identify that. Similar scenarios can be found in English translations *Kamala Devi Chattopadhyay spoke at this meeting, Dr. Ann.* (expected:1.3177, predicted:-0.2999) and *The Warrior Falls are stone's, halting, heraldry and stonework rather than cottages. The cathedral manor is navigable places* (expected:0.1677, predicted:-0.7587). It is clear that the presence of the named entities seem to confuse the algorithm we used, hence it needs to handle named entities in a proper way.

5 Conclusion

In this paper we evaluated different settings of *TransQuest* in sentence-level direct quality assessment. We showed that ensemble results with XLM-R-base and XLM-R-large with data augmentation techniques can improve the performance of TransQuest framework.

The official results of the competition show that *TransQuest* won the first place in all the language pairs in Sentence-Level Direct Assessment task. *TransQuest* is the sole winner in En-Zh, Ne-En and Ru-En language pairs and the multilingual track. For the other language pairs (En-De, Ro-En, Et-En and Si-En) it shares the first place with another system, whose results are not statistically different from ours. The full results of the shared task can be seen in Specia et al. (2020).

In the future, we plan to experiment more with the data augmentation settings. We are interested in augmenting the training file with semantically similar sentences to the test set rather than augmenting with random sentence pairs as we did in this paper. As shown in the error analysis in Section 4.4 the future releases of the framework need to handle named entities properly. We also hope to implement *TransQuest* in document level quality estimation too.

References

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

M Fomicheva, L Specia, S Sun, L Yankovskaya, F Blain, F Guzmán, M Fishel, N Aletras, and V Chaudhary. 2020. Unsupervised quality estimation for neural machine translation. *Transactions of the Association for Computational Linguistics*, Vol 8 (2020).

Erick Fonseca, Lisa Yankovskaya, André F. T. Martins, Mark Fishel, and Christian Federmann. 2019. Findings of the WMT 2019 shared tasks on quality estimation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 1–10, Florence, Italy. Association for Computational Linguistics.

Julia Ive, Frédéric Blain, and Lucia Specia. 2018. deepQuest: A framework for neural-based quality estimation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3146–3157, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Fabio Kepler, Jonay Trénous, Marcos Treviso, Miguel Vera, and André F. T. Martins. 2019. OpenKiwi: An open source framework for quality estimation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 117–122, Florence, Italy. Association for Computational Linguistics.

Hyun Kim, Jong-Hyeok Lee, and Seung-Hoon Na. 2017. Predictor-estimator using multilevel task learning with stack propagation for neural quality estimation. In *Proceedings of the Second Conference on Machine Translation*, pages 562–568, Copenhagen, Denmark. Association for Computational Linguistics.

Tharindu Ranasinghe, Constantin Orasan, and Ruslan Mitkov. 2019. Semantic textual similarity with

Siamese neural networks. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 1004–1011, Varna, Bulgaria. INCOMA Ltd.

Tharindu Ranasinghe, Constantin Orasan, and Ruslan Mitkov. 2020. Transquest: Translation quality estimation with cross-lingual transformers. In *Proceedings of the 28th International Conference on Computational Linguistics*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Sebastian Ruder, Ivan Vulić, and Anders Søgaard. 2019. A survey of cross-lingual word embedding models. *J. Artif. Int. Res.*, 65(1):569–630.

Lucia Specia, Frédéric Blain, Marina Fomicheva, Erick Fonseca, Vishrav Chaudhary, Francisco Guzmán, and André FT Martins. 2020. Findings of the wmt 2020 shared task on quality estimation. In *Proceedings of the Fifth Conference on Machine Translation: Shared Task Papers*.

Lucia Specia, Frédéric Blain, Varvara Logacheva, Ramón Astudillo, and André F. T. Martins. 2018. Findings of the WMT 2018 shared task on quality estimation. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 689–709, Belgium, Brussels. Association for Computational Linguistics.

Lucia Specia, Gustavo Paetzold, and Carolina Scarton. 2015. Multi-level translation quality prediction with QuEst++. In *Proceedings of ACL-IJCNLP 2015 System Demonstrations*, pages 115–120, Beijing, China. Association for Computational Linguistics and The Asian Federation of Natural Language Processing.

Lucia Specia, Kashif Shah, Jose G.C. de Souza, and Trevor Cohn. 2013. QuEst - a translation quality estimation framework. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 79–84, Sofia, Bulgaria. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Yige Xu, Xipeng Qiu, Ligao Zhou, and Xuanjing Huang. 2020. Improving bert fine-tuning via self-ensemble and self-distillation. *arXiv preprint arXiv:2002.10345*.

HW-TSC's Participation at WMT 2020 Quality Estimation Shared Task

Minghan Wang[1], **Hao Yang**[1], **Hengchao Shang**[1], **Daimeng Wei**[1], **Jiaxin Guo**[1], **Lizhi Lei**[1], **Ying Qin**[1], **Shimin Tao**[1], **Shiliang Sun**[3], **Yimeng Chen**[1], **Liangyou Li**[2]
[1]Huawei Translation Services Center, Beijing, China
[2]Huawei Noah's Ark Lab, Hong Kong, China
[3]East China Normal University, Shanghai, China
{wangminghan,yanghao30,shanghengchao,weidaimeng,guojiaxin1,
leilizhi,qinying,taoshimin,chenyimeng,liliangyou}@huawei.com
slsun@cs.ecnu.edu.cn

Abstract

This paper presents our work in the WMT 2020 Word and Sentence-Level Post-editing Effort Quality Estimation (QE) Shared Task. Our system follows standard Predictor-Estimator architecture, with a pre-trained Transformer as the Predictor, and specific classifiers and regressors as Estimators. We integrate Bottleneck Adapter Layers in the Predictor to improve the transfer learning efficiency and prevent from over-fitting. At the same time, we jointly train the word- and sentence-level tasks with a unified model with multitask learning. Pseudo-PE assisted QE (PEAQE) is proposed, resulting in significant improvements on the performance. Our submissions achieve competitive result in word/sentence-level sub-tasks for both of En-De/Zh language pairs.

1 Introduction

Quality Estimation (QE) assesses the translation quality of machine translation (MT) system output when ground truth reference is not available (Specia et al., 2013, 2018). For the word-level QE task, participants are required to tag tokens and gaps of the translation output from an unknown MT system with OK and BAD, as well as tokens in the source with the same tags. The result is measured by Matthews Correlation Coefficient (MCC). For the sentence-level task, participants are required to predict the Human Translation Error Rate (HTER) scores of the machine translation outputs and the result is evaluated in terms of the Pearson's correlation coefficient.

Our team participates in some of the sub-tasks in two language pairs (En-De and En-Zh) (Specia et al., 2020). With regard to the En-De track, at word-level, our model achieves the MCC score of 0.5828 on the target side, and 0.5234 on the source side; at sentence-level, our model ranks the first place with a Pearson R score of 0.7583. With regard to the En-Zh track, we only submit the target side of word-level sub-task, and achieves a MCC score of 0.5872.

Our system is implemented with fairseq (Ott et al., 2019) (for En-De track) and THUMT (Zhang et al., 2017) (for En-Zh track). We extend the original Transformer (Vaswani et al., 2017) model to fit the QE task, and leverage transfer learning to fine-tune the model with the task specific dataset (Dai and Le, 2015; Howard and Ruder, 2018; Radford et al., 2018). The contributions of our work are as follows:

- We follow the Predictor-Estimator (Kim and Lee, 2016; Kim et al., 2017; Wang et al., 2018; Li et al., 2018; Kepler et al., 2019) architecture and build a unified QE model based on the standard Transformer MT model.
- Bottleneck Adapter Layers (Houlsby et al., 2019; Yang et al., 2020) are integrated into the model for efficient transfer learning.
- We propose the Pseudo-PE assisted QE (PEAQE) method which effectively improve the performance.

The architecture of our model is shown in Figure 1.

2 Task Description

A more detailed description of the word- and sentence-level QE tasks is given in this section.

2.1 Word-Level

Word-level QE estimates the translation quality by producing a sequence of tags for both source and target. For target sentences, both tokens and gaps are required to be tagged with OK or BAD, while for source sentences, only tokens are tagged

Proceedings of the 5th Conference on Machine Translation (WMT), pages 1056–1061

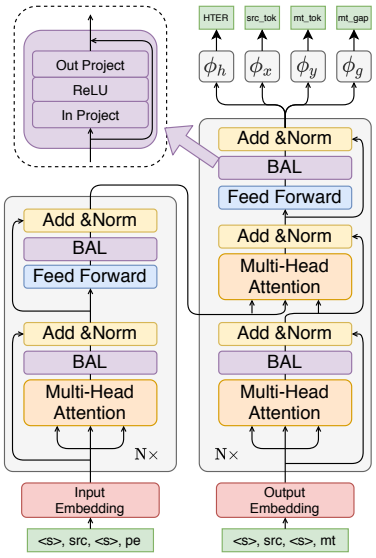


Figure 1: This figure shows the architecture of our model, where SRC and Pseudo-PE are concatenated as the encoder input, a copy of SRC and MT are concatenated as the decoder input. The output feature are passed through four linear layers to make prediction for four tasks.

with OK or BAD. This is usually modelled as a sequential labelling problem. The tag of token indicates whether the word is correctly translated or not, while the tag of gap indicates whether one or more words are missing here. The number of total tags for each MT sentence is $2N + 1$, where N is the number of tokens in the sentence.

The evaluation metrics of the word-level task is the Matthews Correlation Coefficient (MCC), an appropriate measurement for unbalanced labels. MCC is defined as follows:

$$S = \frac{TP + FN}{N}$$
$$P = \frac{TP + FP}{N}$$
$$MCC = \frac{\frac{TP}{N} - SP}{\sqrt{SP(1-S)(1-P)}}, \quad (1)$$

where TP, TN, FP and FN represent for true positives, true negatives, false positives and false positives respectively; and N is the number of instance (Fonseca et al., 2019).

2.2 Sentence-Level

The sentence-level QE predicts the Human Translation Error Rate (HTER) (Specia et al., 2018) of given translation outputs. HTER is an edit-distance measure, calculating the ratio between the number of edits (insertions/deletions/replacements) required and the reference translation length, namely HTER = (Insertions + Delations + Replacement) / Reference Words. In the QE task, where references are not available, HTER is roughly an estimation. As HTER is a real value ranging from 0 to 1, it can be modeled as a regression task. The evaluation metrics of the sentence-level task is the Pearson correlation coefficient.

Attributes	En-De	En-Zh
# Instance	7,000	7,000
# SRC Token	11,4980	115,585
# MT Token	112,342	120,015
% MT Token BAD	28.15	54.33
% MT Gap BAD	4.60	8.04
% SRC Token BAD	26.95	53.60
BLEU (MT, PE)	49.40	30.40
μ(HTER)	0.3181	0.6280
σ(HTER)	0.2017	0.2040

Table 1: The statistics of the training set for both language pairs.

3 Dataset

The dataset contains 7,000 sentences for the training set, 1,000 for the dev and 1,000 for the test. Except from tags and HTER scores (labels), the dataset also provides post-edit (PE) text, as the reference for generating QE labels. Note that this data is also used in the Automatic Post Editing task in WMT 2020. Detailed statistics of the dataset is listed in Table 1, with some metrics of the source (SRC) and translation (MT). The proportion of BAD tags agains OK tags is imbalanced, especially for Gap tags.

Apart from the brief descriptive statistics listed in the table, our in-depth investigation on the provided dataset unveils some interesting findings:

- Different from the dataset in WMT 2019 QE task (Fonseca et al., 2019), which is sampled from IT domain, the dataset this year is collected from Wikipedia. Therefore, mixing data from previous years may not help to improve this year's performance.

- The BLEU score (Papineni et al., 2002) for 2020 dataset is significantly lower than that of 2019, indicating much more operations are required to edit the translation outputs into the references. As a result, the distribution

of labels for 2020 dataset is changed as well when comparing with that of last year.

Unlike a standard translation task, where various data augmentation techniques, such as back-translation (Sennrich et al., 2016), are available, QE task can hardly be improved with data augmentation, as it requires unbiased and high-quality PEs to generate tags and HTER scores. Meanwhile, the change of dataset domain makes it impossible to enlarge the dataset by incorporating the dataset of last year. Facing this challenging task, we propose the PEAQE method, which will be further explained in details in the following section.

4 Model

4.1 Unified QE Model

Our model follows the Predictor-Estimator (Kim et al., 2017; Kepler et al., 2019) architecture. The Predictor is considered as a feature extractor, which can be a pre-trained language model (LM) or a translation model. In our implementation, we use the standard Transformer without the causal mask as the Predictor, which is pre-trained with dataset in news translation task of WMT 2019 En-De and WMT 2020 En-Zh. The Estimator can be task specific classifiers which map the extracted features into the target space, and can be modelled by several fully connected layers. We use a unified QE model to solve both word- and sentence-level tasks by building three classifiers and a regressor to make prediction on SRC tag, MT token tag, MT gap tag and HTER score, respectively.

We define the encoder and decoder of the Transformer as functions f and g; SRC and MT text as X and Y; tags of SRC, MT token and MT gap as V_x, V_y, V_g; and HTER score as V_h. The representation $\mathbf{H}_X$ and $\mathbf{H}_Y$ are obtained by passing the X and Y into the encoder and decoder respectively:

$$\mathbf{H}_X = f(X) \tag{2}$$

$$\mathbf{H}_Y = g(Y, \mathbf{H}_X). \tag{3}$$

For a translation model, we usually append and prepend the special token $\langle s \rangle$ to the SRC and TGT text. Here we follow the same rule and thereby the lengths of SRC and MT embeddings are $M+1$ and $N+1$ respectively. Meanwhile, we append and prepend a special label $\langle pad \rangle$ to the original label sequence during training, but loss of the padded label is not computed. All predictions are obtained by performing specific transformations $\phi_\cdot$ on the hidden stats:

$$\hat{V}_x = \phi_x(\mathbf{H}_X) \tag{4}$$

$$\hat{V}_y = \phi_y(\mathbf{H}_Y) \tag{5}$$

$$\hat{V}_g = \phi_g(\mathbf{H}_Y) \tag{6}$$

$$\hat{V}_h = \phi_h(h_{Y,0}). \tag{7}$$

Note that the regressor ϕ_h only takes the embedding of the MT's first token to make predictions, similar to the usage of [CLS] in BERT (Devlin et al., 2018).

For all classification tasks, we use weighted cross entropy as the loss function, and the weight is calculated as follows: $w_i = \frac{\sum |C_i|}{|C_i|}$, which is the inverse of the proportion of the instance with class C_i. We use weighted cross entropy because labels are highly imbalanced, and the loss should be manipulated with the weight. For sentence-level QE, we use mean squared error (MSE) as the loss function, which is quite intuitive.

The model is trained under the multi-task learning framework by summing up the loss of all subtasks with specific weights:

$$\begin{aligned}\mathcal{L} =& \lambda_h \sqrt{(\hat{V}_h - V_h)^2} \\ &- \sum_{\tau \in \{x,y,g\}} \lambda_\tau \log P(V_\tau | X, Y),\end{aligned} \tag{8}$$

where $\{x, y, g\}$ represents for classification tasks and h represents for regression task, and λ is the weight of loss for a specific task. Although the multi-task setting could improve the overall performance, the evaluation is performed separately, it means we can train models that are optimized for the specific task, which can be achieved by giving larger weight to the loss of that task.

4.2 Bottleneck Adapter Layer

As mentioned in the previous section, the provided training set is relatively small, make the model to be easily over-fitted if all weights are updated. Therefore, we decide to integrate the Bottleneck Adapter Layers (BAL) (Houlsby et al., 2019) while keeping parameters of original Transformer fixed (Yang et al., 2020).

BAL can be easily implemented with two fully-connected layers with a non-linear activation, and is embedded into the Transformer with residual connections after the self-attention layer and the FFN layer, respectively.

In the experiment, we find that the bottle with a "thick" neck ("like FFN layers in the Transformer with higher dimension in the middle part") could further improve the performance without seriously sacrificing training efficiency. More specifically, we tested three neck sizes, i.e. thin, same and thick. The thin and same neck basically reaches 97%-99% of performance compared with training the full Transformer without using BAL, which yields the same result with (Houlsby et al., 2019). By increasing the parameter size of BALs, we find that the performance also increases linearly, finally reaching the pick of 104% of the baseline performance with the neck to have 2 $\times$ hidden size.

4.3 Pseudo-PE Assisted QE

QE tags can be generated with post-edits (PEs) or reference (REF) of MT. In this dataset, PE is provided, and QE tags are generated accordingly, if PE can be directly used to assist the model learning QE tags, the training efficiency will be dramatically increased. Inspired by the Pseudo-PE technique proposed in the (Kepler et al., 2019), we hope to fully leverage PE, for example, integrating them as part of the network input. However, for the test set, where PEs are not available, we must find an alternative approach. So, we made following assumption:

$$\delta(\text{MT}, \text{REF}) \approx \delta(\text{MT}, \text{PE}) + \delta(\text{PE}, \text{REF}), \quad (9)$$

where δ is any distance measurement function. In the equation, PE is regarded as an intermediate node between MT and REF. Under such assumption, if we could find any translation that is better than MT, although not as good as PE, the translation can also be used as a substitute of PE, denoted as PE'. we call this method as Pseudo-PE assisted QE (PEAQE). Finding PE' is relatively easy when we could access unconstrained resources. Using an APE system or a robust online translation system to produce better translation outputs are two feasible approaches. After comparing the BLEU scores of the training set between many online translation services and an APE system trained by us, we decide to use Google Translate outputs as the Pseudo-PE. The BLEU score for official MTs and Google MTs in the dev set are 50.9/ 67.8 for En-De, and 22.62/41.77 for En-Zh, indicating that Google MT outputs, with a high quality, could be used as Pseudo-PEs in the testing phase.

To leverage PEs, we simply concatenate them with the SRCs and encoded them via an encoder. We find that using the features of SRC text from the encoder could not produce acceptable predictions. Therefore, we decide to concatenate SRCs with MTs again on the decoder side, and use the decoder to extract features for both of them. More formally:

$$\mathbf{H}_{[X;Z]} = f([X; Z]) \quad (10)$$

$$\mathbf{H}_{[X;Y]} = g([X; Y], \mathbf{H}_{[X;Z]}), \quad (11)$$

where Z represents for official PEs (training) or Pseudo-PEs (testing). Finally, the hidden state $\mathbf{H}_{[X;Y]}$ is sliced with the max length of X, and recover back to $\mathbf{H}_X$ and $\mathbf{H}_Y$, which are used as in the original model. Official PE and Pseudo-PE can be used respectively during training and testing to assist the model to make better prediction.

5 Experiment

Our experiments of all sub-tasks for En-De and part of sub-tasks for En-Zh trak are performed on the WMT 2020 dataset. The model without Pseudo-PE assistance is considered as the baseline.

5.1 Experimental Settings

Our models are implemented with fairseq (Ott et al., 2019) and THUMT (Zhang et al., 2017). The fairseq version mainly deals with En-De tasks thanks to the pre-trained models trained in WMT 2019 news translation task. The En-Zh pre-trained model is implemented with THUMT and is trained in WMT 2020 news translation task by our team. For the En-De model, input and output embeddings are shared, therefore SRC and TGT text can be conveniently concatenated. For the En-Zh model, vocabulary is not shared, when creating the input sequence, we firstly pass tokens of English (SRC) and Chinese (MT and PE) with specific word embedding layer respectively, and than, concatenate the hidden states of them accordingly. The number of parameters of the En-De and En-Zh models are 270M and 353M, respectively. The batch size used for training is 32. We use Adam (Kingma and Ba, 2015) to optimize parameters with learning rate of 1e-4 without any scheduler. Note that when dealing with labels of sub-tokens, for each token, we only assign the first sub-token with the label and subsequent sub-tokens are assigned with the dummy pad labels, which keeps the distribution of labels unchanged. Our QE models are trained on a Nvidia Tesla V100 GPU, and converge within 5 epochs.

Lang	Model	MCC-MT	MCC-SRC	Pearson-R
En-De (Dev)	Baseline	44.50	32.46	55.26
	+ PEAQE	60.05	45.31	71.69
	+ Ensemble (14)	64.70	51.17	73.33
En-De (Test)	+ Ensemble (14)	58.28	52.34	75.83
En-Zh (Dev)	Baseline	43.06	-	-
	+ PEAQE	57.90	-	-
	+ Ensemble (5)	59.28	-	-
En-Zh (Test)	+ Ensemble (5)	58.72	-	-

Table 2: The experimental results of our model, where the baseline model is introduced in section 4.1. The evaluation results of the test set are from the official leader-board.

5.2 Experimental Results

Table 2 shows the experimental results on the dev and test sets. The performance of the baseline model is relatively poor. By leveraging PEAQE, the model achieves much better performance, demonstrating that integrating PE directly into the QE model could effectively assist the prediction. With PEs, the model can receive stronger supervision signal and is actually learning the procedure done by the tagging script, making the entire learning process easier. However, we clearly understand that the performance of PEAQE strongly depends on the quality of Pesudo-PEs, which becomes another problem that should be solved in the future.

Here is another interesting finding during our experiment. Initially, we also performed experiments with mBERT (Devlin et al., 2018) and XLM (Conneau and Lample, 2019) but not producing desirable results. The reason might be size of the dataset. We find that performing transfer learning with pre-trained NMT model on the limited size QE dataset is more effective than other pre-trained multilingual LMs. We consider that NMT models are naturally fit for MT related tasks because of the learned prior between bilingual text, which might not be captured by multilingual LMs where text in different languages are trained independently.

6 Conclusion

We present our works for WMT 2020 QE shared task. The experimental results demonstrate that performing transfer learning with a pre-trained NMT model on the QE task is effective. Compared to only using SRC and MT text, we propose PEAQE which could significantly improve the performance of the model. But generating reliable Pseudo-PEs that are compatible with QE tasks remains a problem that would be investigated in our future works.

References

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 7057–7067.

Andrew M Dai and Quoc V Le. 2015. Semi-supervised sequence learning. In *Advances in neural information processing systems*, pages 3079–3087.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Erick R. Fonseca, Lisa Yankovskaya, André F. T. Martins, Mark Fishel, and Christian Federmann. 2019. Findings of the WMT 2019 shared tasks on quality estimation. In *Proceedings of the Fourth Conference on Machine Translation, WMT 2019, Florence, Italy, August 1-2, 2019 - Volume 3: Shared Task Papers, Day 2*, pages 1–10.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. *arXiv preprint arXiv:1902.00751*.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*.

Fabio Kepler, Jonay Trénous, Marcos V. Treviso, Miguel Vera, António Góis, M. Amin Farajian, António V. Lopes, and André F. T. Martins. 2019. Unbabel's participation in the WMT19 translation quality estimation shared task. In *Proceedings of the Fourth Conference on Machine Translation, WMT*

2019, Florence, Italy, August 1-2, 2019 - Volume 3: Shared Task Papers, Day 2, pages 78–84.

Hyun Kim and Jong-Hyeok Lee. 2016. Recurrent neural network based translation quality estimation. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 787–792.

Hyun Kim, Jong-Hyeok Lee, and Seung-Hoon Na. 2017. Predictor-estimator using multilevel task learning with stack propagation for neural quality estimation. In *Proceedings of the Second Conference on Machine Translation*, pages 562–568.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Maoxi Li, Qingyu Xiang, Zhiming Chen, and Mingwen Wang. 2018. A unified neural network for quality estimation of machine translation. *IEICE TRANSACTIONS on Information and Systems*, 101(9):2417–2421.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA*, pages 311–318.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*.

Lucia Specia, Frédéric Blain, Marina Fomicheva, Erick Fonseca, Vishrav Chaudhary, Francisco Guzmán, and André FT Martins. 2020. Findings of the wmt 2020 shared task on quality estimation. In *Proceedings of the Fifth Conference on Machine Translation: Shared Task Papers*.

Lucia Specia, Carolina Scarton, and Gustavo Henrique Paetzold. 2018. *Quality Estimation for Machine Translation*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers.

Lucia Specia, Kashif Shah, José GC De Souza, and Trevor Cohn. 2013. QuEst-A translation quality estimation framework. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 79–84.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, \Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Jiayi Wang, Kai Fan, Bo Li, Fengming Zhou, Boxing Chen, Yangbin Shi, and Luo Si. 2018. Alibaba submission for WMT18 quality estimation task. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 809–815.

Hao Yang, Minghan Wang, Ning Xie, Ying Qin, and Yao Deng. 2020. Efficient transfer learning for quality estimation with bottleneck adapter layer. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation, EAMT 2020, Lisbon, Portugal, 3 - 5 November, 2020*, pages 29–34.

Jiacheng Zhang, Yanzhuo Ding, Shiqi Shen, Yong Cheng, Maosong Sun, Huan-Bo Luan, and Yang Liu. 2017. THUMT: an open source toolkit for neural machine translation. *CoRR*, abs/1706.06415.

Tencent submission for WMT20 Quality Estimation Shared Task

Haijiang Wu Zixuan Wang Qingsong Ma Xinjie Wen
Ruichen Wang Xiaoli Wang Yulin Zhang Zhipeng Yao Siyao Peng
PCG & CSIG, Tencent Inc, China
{harywu, zackiewang, qingsongma, jasonxjwen, ruichenwang,
evexlwang, elwinzhang, neokevinyao, logansypeng}@tencent.com

Abstract

This paper presents Tencent's submission to the WMT20 Quality Estimation (QE) Shared Task: Sentence-Level Post-editing Effort for English-Chinese in Task 2. Our system ensembles two architectures, XLM-based and Transformer-based Predictor-Estimator models. For the XLM-based Predictor-Estimator architecture, the predictor produces two types of contextualized token representations, i.e., masked XLM and non-masked XLM; the LSTM-estimator and Transformer-estimator employ two effective strategies, top-K and multi-head attention, to enhance the sentence feature representation. For Transformer-based Predictor-Estimator architecture, we improve a top-performing model by conducting three modifications: using multi-decoding in machine translation module, creating a new model by replacing the transformer-based predictor with XLM-based predictor, and finally integrating two models by a weighted average. Our submission achieves a Pearson correlation of 0.664, ranking first (tied) on English-Chinese (Specia et al., 2020).

1 Introduction

The development of Machine Translation (MT) requires efficient quality evaluation of the outputs. The widely used MT metric BLEU (Papineni et al., 2002) satisfies this demand. However, BLEU requires human reference translations which costs labor and time to generate. Quality Estimation (QE) is an alternative to evaluate the quality of MT outputs with no access to reference translations (Fonseca et al., 2019; Yang et al., 2019).

We participate in the sentence-level task in Task 2 of the WMT20 QE Shared Task for English–Chinese (Specia et al., 2020). The sentence-level task aims to predict the Human-targeted Translation Edit Rate (HTER) (Snover et al., 2006) of the MT output, which reflects the minimal amount of edits that is needed to post-edit the MT output to an acceptable one, thus denotes the overall quality of the MT output.

The classical baseline model QuEst++ (Specia et al., 2015) constructed rule-based features and employed machine learning algorithm to predict HTER scores. Recent neural networks applied the newly-emerged predictor-estimator architecture to QE tasks. Kim et al. (2017) proposed the first predictor-estimator model to extract feature vectors by incorporating large parallel data into a bilingual RNN model, which is subsequently fed into another bidirectional RNN model to predict QE scores. Later on, Fan et al. (2019) replaced the RNN-based predictor by a bidirectional Transformer and added 4-dimensional mis-matching features; Wang et al. (2019) used a Transformer-DLCL based predictor; and Kepler et al. (2019a) introduced BERT and XLM pretrained predictors into their system. Besides the improvements on model architectures, choosing the top-performing models using ensemble techniques further improves the QE performance. For example, the submission using ensemble techniques achieved the best result in the sentence-level QE sub-task in both WMT19 (Fonseca et al., 2019) and CCMT19 (Yang et al., 2019).

We submit a predictor-estimator based QE system, which extends the open-source OpenKiwi framework[1] (Kepler et al., 2019b) to take advantage of recently proposed pre-trained models by transfer learning technique. Our contributions are as follow:

- We propose XLM-based Predictor-Estimator architecture, which introduces the cross-lingual language model (XLM) (Lample and Conneau, 2019) to QE task via transfer learning technique. Instead of directly using target

[1] https://github.com/Unbabel/OpenKiwi

Proceedings of the 5th Conference on Machine Translation (WMT), pages 1062–1067
Online, November 19–20, 2020.

word representations produced by XLM as the predictor output, we propose non-masked XLM representation and masked XLM representation, and adopt further computation to enhance the feature extraction ability.

- We implement LSTM-based estimator and Transformer-based estimator, with two novel strategies to enhance the sentence feature representation, i.e. top-K strategy and multi-head attention strategy.

- We reform Transformer-based Predictor-Estimator (Fan et al., 2019) by using multi-decoding during the machine translation module. Besides, we create a new model by replacing the transformer-based predictor with XLM-based predictor, and then integrate the two models by weighted average.

- We ensemble several single-models by regression algorithms to produce a single sentence-level prediction, which outperforms the commonly-used arithmetic average.

We next describe the models, experiments and results in detail.

2 Models

Our models employ predictor-estimator architecture and OpenKiwi framework. Overall, we implement two predictor-estimator architectures, namely XLM-based Predictor-Estimator and Transformer-based Predictor-Estimator, and ensemble multiple systems to boost performance.

2.1 XLM-based Predictor-Estimator

XLM achieved state-of-the-art performances on several NLP tasks (Lample and Conneau, 2019). We extend XLM by transferring the language model to QE task and proposing a novel XLM-based Predictor-Estimator model.

2.1.1 Predictor

For predictor, we fine-tune XLM with both Masked Language Modeling (MLM) task and Translation Language Modeling (TLM) task using large-scale parallel data following the XLM instructions.[2]

[2]https://github.com/facebookresearch/XLM

XLM representations Instead of using target word representation produced by the fine-tuned XLM as the predictor output as in Kepler et al. (2019a), we propose non-masked XLM representation and masked XLM representation, and adopt further computation to enhance the feature extraction ability. For non-masked XLM, all words are fed into the XLM to predict each word's representation, letting the word itself help to predict its representation. For masked XLM, one target word is masked one time so that the prediction of the masked word leverages information only from the surrounding words and the source context, without any prior information from itself.

Let the length of the target sentence be N, the masking process is repeated N times and then all target word representations are generated. We consider two aspects that influence word representation: the weight of each dimension in the word representation and the language embeddings. Formula (1) describes the final word representation, which is then fed into the estimator as input features to predict HTER scores.

$$Rep_i = R_i \cdot (W_i + Emb_{lang}) \quad (1)$$

In formula (1), i refers to the i-th word in the target sentence and R_i refers to the original representation of the i-th word. W_i denotes the weights of each dimension for the i-th word and Emb_{lang} denotes the language embedding of the target sentence. Rep_i is the final representation of the i-th word.

2.1.2 Estimator

Estimator takes features produced by predictor as the input to predict sentence-level scores of MT outputs. We implement a multi-layer LSTM-estimator and a Transformer-estimator, both of which adopt state-of-the-art strategies to optimize the sentence features.

The last state or the the mean pooling of hidden states are usually taken as the sentence representation. However, they both have weaknesses: the last state losses certain information of the whole sentence due to the information decay problem, while the mean pooling distributes the same weights to all hidden Actually, the contribution of each word to the sentence features varies, which inspires us to take the concept of weight into consideration. We propose two strategies, top-K strategy and multi-head attention strategy to optimize weights from

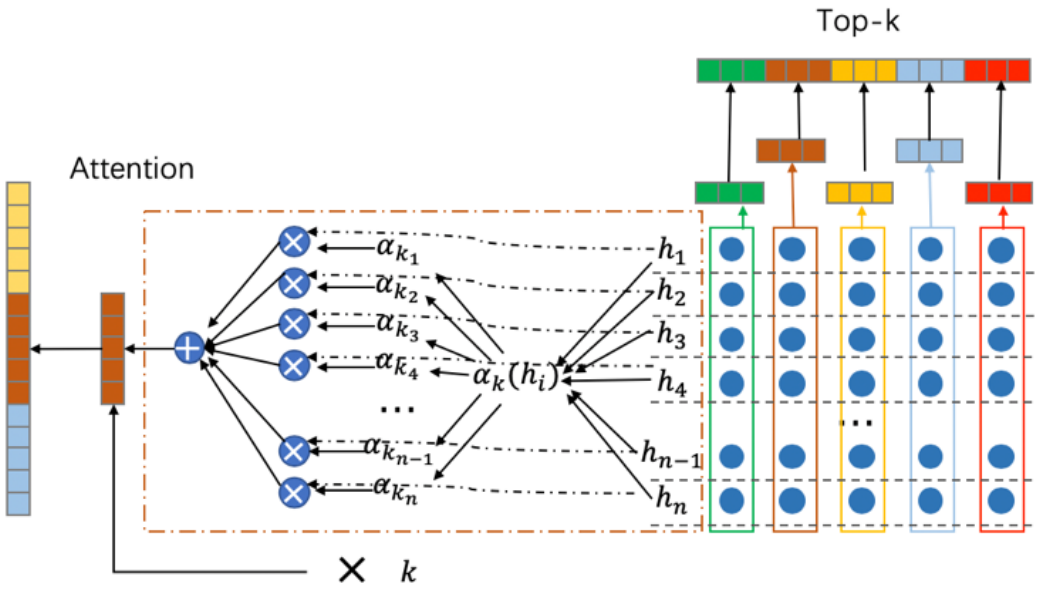


Figure 1: Top-K strategy and Multi-head attention strategy illustration.

two different perspectives, as shown in Figure 1.

Top-K Strategy The hidden state of each word is a vector, and each element of the vector represents one feature dimension. The top-K strategy forms sentence features by concatenating top K elements of each of N feature dimension, creating a vector of size K * N.

Multi-head Attention Strategy Different from Top-K strategy, multi-head attention strategy considers the impact of each word on the sentence features via attention mechanism. For each head, we obtain a vector which is a weighted sum of all the word features. By repeating K times, the final sentence feature is a vector with size K * N. We demonstrate the computation process as in Formula (2) and (3),

$$a_{k_i} = softmax(h_i * W_k), \tag{2}$$

$$f_{sent} = [\sum_i a_{1_i} * h_i, \ldots, \sum_i a_{k_i} * h_i] \tag{3}$$

where a_{k_i} is attention results of each word (h_i), and f_{sent} is the final sentence feature representation.

2.2 Transformer-based Predictor-Estimator

Transformer-based Predictor-Estimator architecture has been proved effective by Fan et al. (2019). Our predictor follows their bidirectional transformer, which contains three modules: self-attention for the source sentence, forward and backward self-attention encoders for the target sentence, and the re-constructor for the target sentence. We include semantic features extracted by bidirectional transformer and human-crafted mismatching features in the model. Our Transformer-based model has three main improvements:

- For transformer-based predictor, we use multi-decoding during the machine translation module.
- We create a XLM-based predictor, which simply replace the predictor part by XLM.
- We take the weighted average of the two models as the final sentence-level prediction as shown in Formula (4). We set α to be 0.8 since the transformer-based predictor contributes more than the XLM-based predictor.

$$Score = \alpha * Score_{Transformer} + (1 - \alpha) * Score_{XLM} \tag{4}$$

2.3 Ensemble

To boost performance, we ensemble several systems to produce a single sentence score prediction. Model stacking (Wolpert, 1992; Breiman, 1996) is an efficient ensemble method in which the predictions, generated by using various single systems, are used as inputs in a second-layer regression algorithm. To avoid over-fitting, we use k-fold cross validation with k = 5 (Martins et al., 2017).

We implement and compare several regression algothrims, i.e., Powell's method (Powell, 1964), Quantile Regression, Support Vector Regression (SVR), and Logistic Regression (LR) to optimize the task on Pearson correlation.

3 Experiments and Results

We conducted three sets of experiments on the WMT20 QE English-Chinese Sentence-level Task in Task 2.

3.1 Dataset

The dataset consists of parallel data between English and Chinese, as well as QE triplets with source texts, target translations and HTER scores. The parallel data is used to train the predictor to produce contextualized features. Specifically, we sampled 45M English-Chinese parallel sentences to train the XLM-based Predictor. For Transformer-based Predictor, we combined the subset of 8.9M parallel sentences in CCMT20 with a set of 15K pseudo data constructed by augmenting the number of entities within the sentences.

3.2 Experiments

3.2.1 Experiments with XLM-based Predictor-Estimator

We experiment with non-masked ($non\text{-}masked$) and masked XLM ($masked$) predictors. We also try to concatenate feature vectors produced by two predictors ($Both$) as the input of the next estimator procedure. For every predictor, we conduct experiments with LSTM-estimator ($LSTM$) and Transformer-estimator (TF), each of which adopts multi-head attention strategy ($attn$) or top-K strategies ($topK$) to improve the sentence representation.

The results in Table 1 show that our QE systems with XLM predictor achieve strong correlation with HTER scores in general. The model with both predictors, LSTM-estimator and multi-head attention strategy (*Both_LSTM_attn*) ranks top with a Pearson score of .6348.

Model	Pearson
Both_LSTM_attn	**.6348**
Both_LSTM_topK	.6244
Both_TF_attn	.6218
Both_TF_topK	.6276
masked_LSTM_attn	.6232
masked_LSTM_topK	6156
masked_TF_attn	.6143
masked_TF_topK	.6260
non-masked_LSTM_attn	.6142
non-masked_LSTM_topK	.6216
non-masked_TF_attn	.6234
non-masked_TF_topK	.6268

Table 1: Pearson correlations of single QE systems with XLM-based Predictor-Estimator on WMT20 English-Chinese development set for sentence-level task.

3.2.2 Experiments with Transformer-based Predictor-Estimator

We extend Transformer-based predictor-estimator (Fan et al., 2019) with the following modifications: we use multi-decoding during Transformer-based predictor, replace Transformer-based predictor with XLM-based predictor to form a new model, and then integrate the two models into one by weighted average with more weights on the Transformer-based predictor.

Table 2 presents the key configurations and results in Transformer-based experiments. Among the four models, Models 1–3 integrate XLM-based estimators into the architecture and achieve the highest Pearson correlations of .646–.647. These integrated models vary in two configurations: whether or not the XLM-estimator has been fine-tuned and whether or not to include source information. We conclude that XLM-based model helps improve Transformer-based Predictor-Estimator performance.

3.2.3 Experiments with ensemble methods

We conduct multiple single QE systems through different model architectures or the same architecture with different parameters. Specifically, we include predictions from 24 XLM-based and 5 Transformer-based Predictor-Estimator systems, and stack them using 4 regressors: Powell's, Quan-

	Transformer	XLM Estimator			Pearson
		Included?	Finetuning?	Input	
Model 1	✔	✔	✔	source & target	**.646**
Model 2	✔	✔	✔	target only	**.647**
Model 3	✔	✔	✘	target only	**.647**
Model 4	✔	✘	N/A	N/A	.633

Table 2: Pearson correlations of single QE systems with Transformer-based Predictor-Estimator on WMT20 English-Chinese development set for sentence-level task.

tile Regression, SVR and LR.

Results in Table 3 prove the effectiveness of ensemble techniques, when compared to results shown in Tables 1 and 2. We also conclude that regression algorithms outperform simple averaging ("Average" in Table 3), and among them, Logistic Regression achieves the best Pearson score of .6785.

Ensemble methods	**Pearson**
Average	.6521
Powell's	.6515
Quantile Regression	.6699
Support Vector Regression	.6735
Logistic Regression	**.6785**

Table 3: Pearson correlations of ensemble QE systems on WMT20 English-Chinese development set for sentence-level task.

4 Conclusion

We describe Tencent's submission to the WMT20 Quality Estimation sentence-level task in task 2. Our systems are based on predictor-estimator architecture and built upon OpenKiwi framework. We implement two predictor-estimator architectures, XLM-based Predictor-Estimator and Transformer-based Predictor-Estimator. For XLM-based Predictor-Estimator, we produces two kinds of contextual token representation, masked and non-masked representations. Both LSTM-estimator and Transformer-estimator are conducted to predict the MT output scores by using the features produced from the predictors. Top-K strategy and multi-head attention strategy are employed to enhance the sentence feature representation. For Transformer-based Predictor-Estimator, we integrate one model based on XLM-based predictor to enhance the overall performance. Stacking ensemble is also proved to be more effective than simple averaging integration.

References

Leo Breiman. 1996. Stacked regressions. *Machine learning*, 24(1):49–64.

Kai Fan, Jiayi Wang, Bo Li, Fengming Zhou, Boxing Chen, and Luo Si. 2019. "bilingual expert" can find translation errors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6367–6374.

Erick Fonseca, Lisa Yankovskaya, André FT Martins, Mark Fishel, and Christian Federmann. 2019. Findings of the wmt 2019 shared tasks on quality estimation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 1–10.

Fábio Kepler, Jonay Trénous, Marcos Treviso, Miguel Vera, António Góis, M Amin Farajian, António V Lopes, and André FT Martins. 2019a. Unbabel's participation in the wmt19 translation quality estimation shared task. *arXiv preprint arXiv:1907.10352*.

Fábio Kepler, Jonay Trénous, Marcos Treviso, Miguel Vera, and André FT Martins. 2019b. Openkiwi: An open source framework for quality estimation. *arXiv preprint arXiv:1902.08646*.

Hyun Kim, Hun-Young Jung, Hongseok Kwon, Jong-Hyeok Lee, and Seung-Hoon Na. 2017. Predictor-estimator: Neural quality estimation based on target word prediction for machine translation. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 17(1):1–22.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *arXiv preprint arXiv:1901.07291*.

André FT Martins, Marcin Junczys-Dowmunt, Fabio N Kepler, Ramón Astudillo, Chris Hokamp, and Roman Grundkiewicz. 2017. Pushing the limits of translation quality estimation. *Transactions of the Association for Computational Linguistics*, 5:205–218.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the*

40th annual meeting of the Association for Computational Linguistics, pages 311–318.

Michael JD Powell. 1964. An efficient method for finding the minimum of a function of several variables without calculating derivatives. *The computer journal*, 7(2):155–162.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of association for machine translation in the Americas*, volume 200. Cambridge, MA.

Lucia Specia, Frédéric Blain, Marina Fomicheva, Erick Fonseca, Vishrav Chaudhary, Francisco Guzmán, and André FT Martins. 2020. Findings of the wmt 2020 shared task on quality estimation. In *Proceedings of the Fifth Conference on Machine Translation: Shared Task Papers*.

Lucia Specia, Gustavo Paetzold, and Carolina Scarton. 2015. Multi-level translation quality prediction with quest++. In *Proceedings of ACL-IJCNLP 2015 System Demonstrations*, pages 115–120.

Ziyang Wang, Hui Liu, Hexuan Chen, Kai Feng, Zeyang Wang, Bei Li, Chen Xu, Tong Xiao, and Jingbo Zhu. 2019. Niutrans submission for ccmt19 quality estimation task. In *China Conference on Machine Translation*, pages 82–92. Springer.

David H Wolpert. 1992. Stacked generalization. *Neural networks*, 5(2):241–259.

Muyun Yang, Xixin Hu, Hao Xiong, Jiayi Wang, Yiliyaer Jiaermuhamaiti, Zhongjun He, Weihua Luo, and Shujian Huang. 2019. Ccmt 2019 machine translation evaluation report. In *China Conference on Machine Translation*, pages 105–128. Springer.

Zero-Shot Translation Quality Estimation with Explicit Cross-Lingual Patterns

Lei Zhou[§] **Liang Ding**[†] **Koichi Takeda**[§]
[§]FVCRC, Graduate School of Informatics, Nagoya University
{zhou.lei@a.mbox,takedasu@i}.nagoya-u.ac.jp
[†]UBTECH Sydney AI Centre, School of Computer Science
Faculty of Engineering, The University of Sydney
ldin3097@uni.sydney.edu.au

Abstract

This paper describes our submission of the WMT 2020 Shared Task on Sentence Level Direct Assessment, Quality Estimation (QE). In this study, we empirically reveal the *mismatching issue* when directly adopting BERTScore (Zhang et al., 2020) to QE. Specifically, there exist lots of mismatching errors between source sentence and translated candidate sentence with token pairwise similarity. In response to this issue, we propose to expose explicit cross lingual patterns, *e.g.* word alignments and generation score, to our proposed zero-shot models. Experiments show that our proposed QE model with explicit cross-lingual patterns could alleviate the mismatching issue, thereby improving the performance. Encouragingly, our zero-shot QE method could achieve comparable performance with supervised QE method, and even outperforms the supervised counterpart on 2 out of 6 directions. We expect our work could shed light on the zero-shot QE model improvement.

1 Introduction

Translation quality estimation (QE) (Blatz et al., 2004; Specia et al., 2018, 2020) aims to predict the quality of translation hypothesis without golden-standard human references, setting it apart from reference-based translation metrics. Existing reference-based evaluation metrics, e.g. BLEU (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005), NIST (Doddington, 2002), ROUGE (Lin, 2004), TER (Snover et al., 2006), are commonly used in language generation tasks including translation, summarization, and captioning but all heavily rely on the quality of given references.

Recently, (Edunov et al., 2020) show that reference-based automatic evaluation metrics, e.g., BLEU, are not always reliable because the human translated references are translationese (Koppel and Ordan, 2011; Graham et al., 2019). Thus, an automatic method with no access to any references, i.e., QE, is highly appreciated.

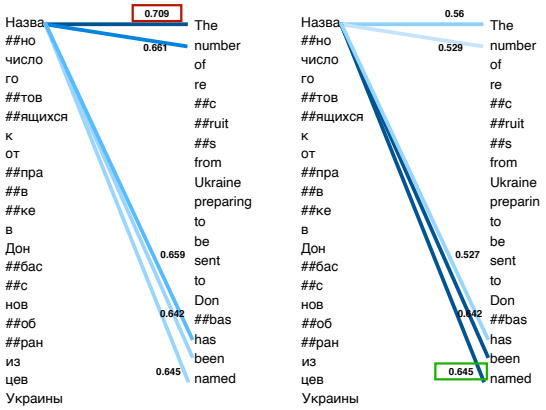


Figure 1: Example of mismatching error, Russian→English. On the left, token "Назва" is mismatched to "The" with the maximal probability (within the red rectangle) only. On the right, guided by our proposed cross-lingual patterns, "Назва" is correctly matched to the token "named" with the maximal probability (within the green rectangle.)

In this paper, we mainly focus on sentence level QE metrics, where existing studies categorize it into two classes: 1) supervised QE with human assessment as supervision signal: a feature extractor stacked with an estimator (Yankovskaya et al., 2019; Wang et al., 2016b; Fan et al., 2019); 2) unsupervised QE without human assessment, which normally based on the pre-trained word embeddings, for example, YISI (Lo, 2019) and BERTScore (Zhang et al., 2020). Our work follows the latter, where we adopt BERTScore (Zhang et al., 2020) without extra fine-tuning. In particular, we implement our approach upon the pre-trained multilingual BERT (Devlin et al., 2019) and XLM (Conneau and Lample, 2019).

We first empirically reveal the *mismatching issue* when directly adopting BERTScore (Zhang

Proceedings of the 5th Conference on Machine Translation (WMT), pages 1068–1074

et al., 2020) to QE task. Specifically, there exist lots of mismatching errors between source tokens and translated candidate tokens when performing greedy matching with pairwise similarity. Figure 1 shows an example of the mismatching error, where the Russian token "Назва" is mismatched to the English token "The" due to lacking of proper guidance.

To alleviate this issue, we design two explicit cross-lingual patterns to augment the BERTScore as a QE metric:

- CROSS-LINGUAL ALIGNMENT MASKING: we design an alignment masking strategy to provide the pairwise similarity matrix with extra guidance. The alignment is derived from GIZA++ (Och and Ney, 2003).
- CROSS-LINGUAL GENERATION SCORE: we obtain the perplexity, dubbed *ppl*, of each target side token by force decoding with a pre-trained cross-lingual model, e.g. multilingual BERT and XLM. This generation score is weighted added on the similarity score.

2 Methods

2.1 BERTScore as Backbone

A pre-trained multilingual model generates contextual embeddings of both source sentence and translated candidate sentence, such that this pair of sentences in different language can be mapped to the same continuous feature space. Given a source sentence $x = \langle x_1, \ldots, x_k \rangle$, the model generates a sequence of vectors $\langle \mathbf{x_1}, \ldots, \mathbf{x_k} \rangle$ while the candidate $\hat{y} = \langle \hat{y}_1, \ldots, \hat{y}_l \rangle$ is mapped to $\langle \hat{\mathbf{y}}_\mathbf{1}, \ldots, \hat{\mathbf{y}}_\mathbf{l} \rangle$. Different from the reference-based BERTScore, where they compute the pairwise similarity between reference sentence and translated candidate sentence, we calculate the pairwise similarity between the source and translated candidate with dot-production, i.e., $\mathbf{x_i}^\top \hat{\mathbf{y}}_\mathbf{j}$. We adopt greedy matching to force each source token to be matched to the most similar target token in the translated candidate sentence. The QE function based on BERTScore backbone therefore can be formulated as:

$$
\begin{aligned}
R_{\text{BERT}} &= \frac{1}{|x|} \sum_{x_i \in x} \max_{\hat{y}_j \in \hat{y}} \mathbf{x_i}^\top \hat{\mathbf{y}}_\mathbf{j}, \\
P_{\text{BERT}} &= \frac{1}{|\hat{y}|} \sum_{\hat{y}_j \in \hat{y}} \max_{x_i \in x} \mathbf{x_i}^\top \hat{\mathbf{y}}_\mathbf{j}, \\
F_{\text{BERT}} &= 2 \frac{P_{\text{BERT}} \cdot R_{\text{BERT}}}{P_{\text{BERT}} + R_{\text{BERT}}}.
\end{aligned}
\tag{1}
$$

where R_{BERT}, P_{BERT} and F_{BERT} are inherited from Zhang et al. (2020), representing Recall rate, Precision rate and F-score, respectively.

2.2 Alignment Masking Strategy

With aforementioned QE function, we can follow Zhang et al. (2020) to obtain the distance between the source sentence and translated candidate sentence via directly adding up the maximum similarity score of each token pair. However, because there exist lots of mismatching errors (as shown in Figure 1), above sentence-level similarity calculation may be sub-optimal. Moreover, Zhang et al. (2020)'s calculation is suitable for monolingual scenario, which may be insensitive for cross-lingual computation. Thus, we propose to augment our QE metric with more cross-lingual signals.

Inspired by Ding et al. (2020), where they show it's possible to augment cross-lingual modeling by leveraging cross-lingual explicit knowledge. we therefore employ word alignment knowledge from external models, e.g., GIZA++[1], as additional information.

Alignment masking Both BERT (Devlin et al., 2019) and XLM (Conneau and Lample, 2019) utilize BPE tokenization (Sennrich et al., 2016). It should be noted that in this paper, by word alignment we mean alignment of BPE tokenized word and subword units. Given a tokenized source sentence x and candidate sentence $\hat{y}$, alignment (Och and Ney, 2003) is defined as a subset of the Cartesian product of position, $\mathcal{A} \subseteq \{(i, j) : i = 1, \ldots, k; j = 1, \ldots, l\}$. Alignment results represented by $\mathcal{M}$ is defined as:

$$
\mathcal{M} = \begin{cases} 1 & (i,j) \in \mathcal{A} \\ 0 \leq a \leq 1 & \text{otherwise} \end{cases}
\tag{2}
$$

$\mathcal{M}$ is a penalty function over the similarity of unaligned tokens. It's a mask like matrix to assign a penalty weight a [2] to the similarity of unaligned tokens while keeping that of aligned ones unchanged, as illustrated in Figure 2. Thus, greedy matching is performed on a renewed similarity matrix, which is defined as the average of $\mathbf{x_i}^\top \hat{\mathbf{y}}_\mathbf{j}$ and masked $\mathbf{x_i}^\top \hat{\mathbf{y}}_\mathbf{j}$ by word alignment. For example, R_{BERT}

[1] https://github.com/moses-smt/giza-pp

[2] In our preliminary studies, $a = 0.8$ picking from $\{0.0, 0.2, 0.4, 0.8, 1.0\}$ performs best, which then leaves as the default setting in the following experiments.

#	Metrics	en-de	en-zh	ro-en	et-en	ne-en	si-en	ru-en
1	Baseline (test)	0.146	0.190	0.685	0.477	0.386	0.374	0.548
2	BERT	**0.120**	0.167	0.650	0.306	0.475	-	**0.354**
3	BERT (align)	0.091	0.170	0.672	0.307	**0.478**	-	0.340
4	BERT (ppl)	0.068	0.187	0.671	0.321	0.468	-	0.311
5	BERT (align+ppl)	0.099	**0.189**	**0.677**	**0.324**	0.477	-	0.332

Table 1: Pearson correlations with sentence-level Direct Assessment (DA) scores. The results of supervised baseline (Kepler et al., 2019), provided by the organizer, show it's agreement with DA scores on the test set of WMT20 QE. As DA scores on test set aren't available at this point, we report our experiment results on valid set.

#	Metrics	en-de	en-zh	ro-en	et-en	ne-en	si-en	ru-en
1	BERT	**0.143**	0.131	0.389	0.217	0.318	-	**0.259**
2	BERT (align)	0.122	0.133	0.422	0.219	**0.322**	-	0.251
3	BERT (ppl)	0.105	0.145	0.416	0.225	0.315	-	0.240
4	BERT (align+ppl)	0.132	**0.152**	**0.439**	**0.228**	0.320	-	0.247

Table 2: Kendall correlations with sentence-level Direct Assessment (DA) scores.

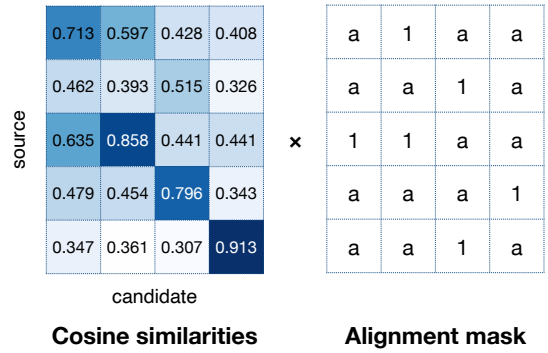


Figure 2: Word alignment as a mask matrix

is changed into:

$$R_{\text{BERT(align)}} = \frac{1}{2|x|}\sum_{x_i \in x} \max_{\hat{y}_j \in \hat{y}} (\mathbf{x_i}^\top \hat{\mathbf{y}}_\mathbf{j} + \mathcal{M} \cdot \mathbf{x_i}^\top \hat{\mathbf{y}}_\mathbf{j}) \tag{3}$$

which can be characterized as balancing our proposed extra explicit cross-lingual patterns, i.e., word alignment.

2.3 Generation Score

In additional to token similarity score, we introduce force-decoding perplexity of each target token as a cross-lingual generation score. For better coordination and considering our cross-lingual setting, we use the same pre-trained cross-lingual model, e.g. multilingual BERT, for both token embedding extraction and masked language model (MLM) perplexity generation. This cross-lingual generation score is added as:

$$F_{\text{BERT(ppl)}} = (1-\lambda) * F_{\text{BERT}} + \lambda * ppl_{\text{MLM}} \tag{4}$$

where the λ can be seen as a variable that regulates the interpolation ratio between F_{BERT} and our proposed ppl_{MLM}, making the generation score after combination more wisely. The effect of λ will be discussed in the experiments.

3 Experimental Results

3.1 Data

Main experiments were conducted on the WMT20 QE Shared Task, Sentence-level Direct Assessment language pairs. The task contains 7 directions, including:

- English→German (**en-de**)
- English→Chinese (**en-zh**)
- Romanian→English (**ro-en**)
- Estonian→English (**et-en**)
- Nepalese→English (**ne-en**)
- Sinhala→English (**si-en**)
- Russian→English (**ru-en**)

Each of them consists of 7K training data, 1K validation data and 1K test data.

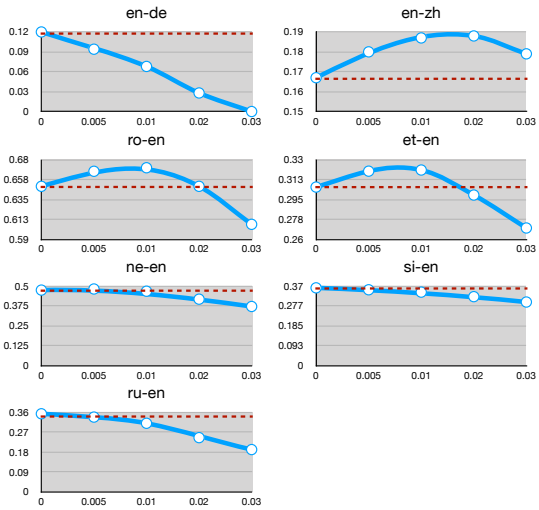


Figure 3: Pearson correlations with Direct Assessment (DA) scores when $\lambda \in [0, 0.03]$.

3.2 Setup

Based on our proposed QE metric in Section 2.1, we conduct the validataion and main experiments with two pre-trained cross-lingual model: bert-base-multilingual-cased[3] (12-layer, 768-hidden, 12-heads, trained on 104 languages) and xlm-mlm-100-1280[4] (16-layer, 1280-hidden, 16-heads, trained on 100 languages) for both contextual embedding representation and generation score. The 9th layer of multilingual BERT and the 11th of XLM are used to generate contextual embedding representations. Furthermore, we obtain bidirectional word alignment of all the training, validation and test dataset with GIZA++. Notably, this work is a zero-shot approach that doesn't involve training on Direct Assessment (DA) scores, which makes our method suitable for real industry scenarios.

3.3 Ablation Study

In order to maximize the advantages of our proposed method for zero-shot translation QE, we conducted extensive ablation studies. We report the results of ablation studies on the validation dataset.

Effect of λ We conduct ablation studies to empirically decide the value of of λ in Equation 4 when introducing generation scores. We observe positive effect of proper weighted additional generation score on **en-zh**, **ro-en**, **et-en**, **ne-en**, **si-en**.

[3] https://huggingface.co/bert-base-multilingual-cased

[4] https://huggingface.co/xlm-mlm-100-1280

	mBERT	XLM
en-de	0.120	0.056
en-zh	0.167	0.008
ro-en	0.650	0.568
et-en	0.306	0.254
ne-en	0.475	0.398
si-en	-	0.362
ru-en	0.354	0.228

Table 3: This is a comparison between multilingual BERT ("mBERT") and XLM in terms of the Pearson correlations with Direct Assessment (DA) scores. Multilingual BERT performs better than XLM.

As illustrated in Figure 3, considering the average performance, we pick $\lambda = 0.01$ from $[0, 0.03]$.

Effect of different pretrained models We also investigated the effect to deploy our proposed fixed cross-lingual patterns on different state-of-the-art large scale pre-trained models, e.g., XLM (Conneau and Lample, 2019) (xlm-mlm-100-1280), BERT (Zhang et al., 2020) (bert-base-multilingual-cased). Table 3 lists a comparison of multilingual BERT and XLM in terms of the Pearson correlations with Direct Assessment (DA) scores. As seen, multilingual BERT outperforms XLM on almost all language pairs, excepting for **si-en**. One possible reason is that multilingual BERT is not pre-trained on Sinhala corpus while XLM does. In this end, we generate our final submission with XLM in **si-en** direction, and with multilingual BERT in other directions.

3.4 Main Results

In the main experiments, we evaluate the agreement of our approach with Direct Assessment (DA) scores on validation dataset, as DA scores of the test set are not available at this point. Baseline results, which are evaluated on test set though, are also listed for general comparison.

As shown in Table 1, our method could achieve improvements on 4 out of 6 directions, including **en-zh**, **ro-en**, **et-en** and **ne-en**. Particularly, combination of two strategies, i.e., CROSS-LINGUAL ALIGNMENT and CROSS-LINGUAL GENERATION SCORE, could achieve better performance on **en-zh**, **ro-en** and **et-en** directions.

Besides Pearson correlations, we also calculated Kendall correlations for all language pairs. As seen in Table 2, the trends of Kendall correlations

	Ours	Kepler et al. (2019)
en-de	0.111	0.146
en-zh	0.085	0.190
ro-en	0.650	0.685
et-en	-	-
ne-en	**0.488**	0.386
si-en	**0.388**	0.374
ru-en	0.400	0.548

Table 4: Comparison of our submission and supervised baseline (Kepler et al., 2019) on WMT20 sentence-level QE official test set, in terms of Pearson correlations.

are same as Pearson correlations, validating the effectiveness of our proposed methods.

3.5 Official Evaluations

The official automatic evaluation results of our submissions for WMT 2020 are presented in Table 4. We participated QE (Sentence-Level Direct Assessment) in following language pairs: **en-de**, **en-zh**, **ro-en**, **ne-en**, **si-en**, **ru-en**, except for **et-en**. From the official evaluation results (Specia et al., 2020) in terms of absolute Pearson Correlation, our submission achieves higher performance than supervised baseline (Kepler et al., 2019) in **ne-en** and **si-en** (As shown in Table 4).

Encouragingly, our proposed zero-shot QE metric could achieve comparable performance with supervised QE method, and even outperforms the supervised counterpart on 2 out of 6 directions.

4 Related Work

MT evaluation Taking sentence-level evaluation as an example, reference-based metrics describe to which extend a candidate sentence is similar to a reference one (Sellam et al., 2020). BLEU (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005), NIST (Doddington, 2002), ROUGE (Lin, 2004) measure such similarity through n-gram matching, which is restricted to the exact form of sentences. TER (Snover et al., 2006) and CHARACTER (Wang et al., 2016b) use edit distance at word or character level to indicate the distance between candidate and reference. Different from these metrics that are restricted to the exact form of sentences, recent dominated neural model metrics *learn* to evaluate with human assessment as supervision signal, such as BEER (Stanojević and Sima'an, 2014) and RUSE (Shimanaka et al., 2018), or others as YiSi (Lo, 2019) and BERTScore (Zhang et al., 2020) , evaluate with pre-trained word embedding, without using human assessment.

Incorporating Explicit Knowledge Several approaches have incorporated pre-defined or learned features into neural networks. Tai et al. (2015) demonstrate that incorporating structured semantic information could enhance the representations. Sennrich and Haddow (2016) feed the encoder cell combined embeddings of linguistic features including lemmas, subword tags, etc. Ding et al. (2017) leverage the domain knowledge to perform data selection to improve the machine translation models. Ding and Tao (2019) incorporate the structure patterns of sentences, i.e., syntax, into the Transformer network to enhance seq2seq modeling performance. Raganato et al. (2020) utilize the pre-defined fixed patterns to replace the attention weights and show promising results. Inspired by above works, we propose to augment zero-shot QE model with cross-lingual patterns.

5 Conclusion and Future Work

In this work, we revealed a mismatching issue in zero-shot QE modeling. To alleviate it, we introduced two explicit cross-lingual patterns based on BERTScore backbone. Extensive experiments indicated that our proposed patterns, without fine-tuning, the QE model can be improved marginally. Notably, our zero-shot QE method outperforms supervised QE model on 2 out of 6 directions, shedding light on zero-shot QE researches.

In the future, we plan to explore more strategies for incorporating various auxiliary information and better in-domain fine-tuning (Gururangan et al., 2020) or introduce an non-autoregressive refiner (Wu et al., 2020) to address our revealed *mismatching issue*. Also, it will be interesting to apply QE metrics on document-level machine translations with considering the dropped pronoun (Wang et al., 2016a, 2018).

Acknowledgments

Lei Zhou is supported by a project commissioned by the New Energy and Industrial Technology Development Organization (NEDO). The authors wish to thank the anonymous reviewers for their insightful comments and suggestions.

References

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the ACL workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*.

John Blatz, Erin Fitzgerald, George Foster, Simona Gandrabur, Cyril Goutte, Alex Kulesza, Alberto Sanchis, and Nicola Ueffing. 2004. Confidence estimation for machine translation. In *COLING*.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *NIPS*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

Liang Ding, Yanqing He, Lei Zhou, and Qingmin Liu. 2017. Combining domain knowledge and deep learning makes nmt more adaptive. In *CWMT*.

Liang Ding and Dacheng Tao. 2019. Recurrent graph syntax encoder for neural machine translation. *arXiv*.

Liang Ding, Longyue Wang, and Dacheng Tao. 2020. Self-attention with cross-lingual position representation. In *ACL*.

George Doddington. 2002. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *HLT*.

Sergey Edunov, Myle Ott, Marc'Aurelio Ranzato, and Michael Auli. 2020. On the evaluation of machine translation systems trained with back-translation. In *ACL*.

Kai Fan, Jiayi Wang, Bo Li, Fengming Zhou, Boxing Chen, and Luo Si. 2019. "Bilingual Expert" Can Find Translation Errors. *AAAI*.

Yvette Graham, Barry Haddow, and Philipp Koehn. 2019. Translationese in machine translation evaluation. *arXiv*.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *ACL*.

Fabio Kepler, Jonay Trénous, Marcos Treviso, Miguel Vera, and André F. T. Martins. 2019. OpenKiwi: An open source framework for quality estimation. In *ACL*.

Moshe Koppel and Noam Ordan. 2011. Translationese and its dialects. In *ACL*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*.

Chi-kiu Lo. 2019. Yisi-a unified semantic mt quality evaluation and estimation metric for languages with different levels of available resources. In *WMT*.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational linguistics*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*.

Alessandro Raganato, Yves Scherrer, and Jörg Tiedemann. 2020. Fixed encoder self-attention patterns in transformer-based machine translation. In *arXiv*.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *ACL*.

Rico Sennrich and Barry Haddow. 2016. Linguistic input features improve neural machine translation. In *WMT*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *ACL*.

Hiroki Shimanaka, Tomoyuki Kajiwara, and Mamoru Komachi. 2018. Ruse: Regressor using sentence embeddings for automatic machine translation evaluation. In *WMT*.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *AMTA*.

Lucia Specia, Frédéric Blain, Marina Fomicheva, Erick Fonseca, Vishrav Chaudhary, Francisco Guzmán, and André FT Martins. 2020. Findings of the wmt 2020 shared task on quality estimation. In *WMT*.

Lucia Specia, Frédéric Blain, Varvara Logacheva, Ramón Astudillo, and André FT Martins. 2018. Findings of the wmt 2018 shared task on quality estimation. In *WMT*.

Miloš Stanojević and Khalil Sima'an. 2014. Beer: Better evaluation as ranking. In *WMT*.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *ACL*.

Longyue Wang, Zhaopeng Tu, Shuming Shi, Tong Zhang, Yvette Graham, and Qun Liu. 2018. Translating pro-drop languages with reconstruction models. In *AAAI*.

Longyue Wang, Zhaopeng Tu, Xiaojun Zhang, Hang Li, Andy Way, and Qun Liu. 2016a. A novel approach to dropped pronoun translation. In *NAACL*.

Weiyue Wang, Jan-Thorsten Peter, Hendrik Rosendahl, and Hermann Ney. 2016b. Character: Translation edit rate on character level. In *WMT*.

Di Wu, Liang Ding, Fan Lu, and J. Xie. 2020. Slotrefine: A fast non-autoregressive model for joint intent detection and slot filling. In *EMNLP*.

E Yankovskaya, A Tättar, M Fishel Volume 3 Shared Task Papers, Day, and 2019. 2019. Quality estimation and translation metrics via pre-trained word and sentence embeddings. In *WMT*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *ICLR*.

NLPRL System for Very Low Resource Supervised Machine Translation

Rupjyoti Baruah, Rajesh Kumar Mundotiya, Amit Kumar, Anil Kumar Singh
Department of Computer Science & Engineering
Indian Institute of Technology (BHU)
Varanasi, India
{rupjyotibaruah.rs.cse18, rajeshkm.rs.cse16}@iitbhu.ac.in
{amitkumar.rs.cse17, aksingh.cse}@iitbhu.ac.in

Abstract

This paper describes the results of the system that we used for the WMT20 very low resource (VLR) supervised MT shared task. For our experiments, we use a byte-level version of BPE, which requires a base vocabulary of size 256 only. BPE based models are a kind of sub-word models. Such models try to address the Out of Vocabulary (OOV) word problem by performing word segmentation so that segments correspond to morphological units. They are also reported to work across different languages, especially similar languages due to their sub-word nature. Based on BLEU cased score, our NLPRL systems ranked ninth for HSB to GER and tenth in GER to HSB translation scenario.

1 Introduction

We report the results for our system that was used for our participation in the WMT20 shared task (Barrault et al., 2019) on very low resource Machine Translation (MT). The MT systems were built for the language pair Upper Sorbian (HSB) and German (GER) in both translation directions.

The Sorbian languages are the West Slavic branch of the Indo-European languages, which have further categorized into two closely related languages, Upper Sorbian and Lower Sorbian. The categories of this language are recognized as a different and distinct language in the European Charter for Regional or Minority languages (Dolowy-Rybinska, 2011). Upper Sorbian is a minority language of Germany that is spoken by $10,000$ to $15,000$ speakers (Elle, 2010), although this number is continually declining (Dołowy-Rybińska, 2018). To counter this, attempts are being made to increase the number of Sorbian speakers through bilingual educational scenarios and MT[1].

[1] https://minorityrights.org/minorities/sorbs/

Low resource MT was being attempted even before Neural Machine Translation (NMT) became the state-of-the-art. Several methods are used to improve the accuracy and quality of the low-resource SMT systems by using comparable corpora (Irvine and Callison-Burch, 2013; Babych et al., 2007), pivot language (English or non-English) technique (Ahmadnia et al., 2017; Paul et al., 2013), and using related resource-rich language (Nakov and Ng, 2012).

We use a byte-level version of Byte Pair Encoding based model with a Transformer for our experiments. The main motivation was to try out this model for the shared task and see how it works under a shared task setting.

2 Background

NMT is an end-to-end learning system (Bahdanau et al., 2015), based on the data-driven approach of machine translation, that requires a massive amount of parallel data for training.

To overcome the lack of such data, several techniques have been tried out which are based on semi-supervised learning (Zheng et al. (2019)), unsupervised learning (Sun et al. (2020)), data augmentation (Siddhant et al. (2020)), transfer learning (Aji (2020)), meta-learning (Li et al. (2020)), pivot-based (Kim et al. (2019)), and multilingual machine translation (Dabre et al. (2020)).

A model-agnostic meta-learning algorithm (Finn et al., 2017) for low-resource NMT exploits the multilingual high-resource language tasks (Gu et al., 2018b). Gu et al. (2018a) achieved significant improvement in performance by utilizing a transfer-learning approach for extremely low resource languages.

Another proposed solution is to use word segmentation units, e.g. characters (Chung et al., 2016), mixed word/characters (Luong and Man-

Proceedings of the 5th Conference on Machine Translation (WMT), pages 1075–1078

ning, 2016), or more intelligent sub-words (Sennrich et al., 2016). It is claimed that an NMT model using such an approach is capable of open-vocabulary translation by encoding rare and unknown words as sequences of sub-word units.

The purpose of our experiments was to try out a supervised NMT system for the low resource language like HSB to GER and vice-versa for the WMT20 shared task.

3 System Description

The standard Transformer architecture proposed by Vaswani et al. (2017) is used for this experiment. This architecture is able to handle long-term dependencies among input tokens, output tokens and between input-output by multi-head attention mechanism. Our method based on the model architecture of Wang et al. (2020), which had used the **B**yte-level BPE (BBPE).

The BBPE encoding is deployed on the Byte Pair Encoding (BPE) (Sennrich et al., 2016), which is a subword algorithm to find a way to represent the given entire text dataset with a small number of tokens. BPE tries to find a balance between character- and word-level hybrid representations, enabling the encoding of any rare words in the vocabulary with appropriate subword tokens without introducing any "unknown" tokens. These segmented byte sequences are encoded into variable-length tokens, i.e., n-grams, which leads to the generation of the BPE vocabulary with byte n-grams. Before being fed to the Transformer model, the learned BBPE passes through bidirectional GRU, which enables to retain contextualization between byte representation of BPE.

4 Experimental Setup

We use the Fairseq[2] (Ott et al., 2019) library to train the Transformer with the same learning rate as in the original paper.

4.1 Dataset and Preprocessing

Our models were trained on the data provided by the Workshop on Machine Translation (WMT) 2020. The statistics about the training, validation and test sets are 60000, 2000 and 2000, respectively for both directional pairs (HSB - GER).

We obtained 1727916 and 1710293 tokens of the GER and HSB, respectively, from the train set for preprocessing. The BPE vocabulary, Byte vocabulary and Character vocabulary are 16384, 2048 and 4096, respectively, for generating binary dataset by using fairseq-preprocess. The BBPE used as a subword BPE tokenizer, where preprocessing was performed using lowercasing only. This is beneficial from the low resource point of view, but it loses the case information for German, which could have affected the results.

4.2 Training Details

We trained the Transformer model with Bi-GRU embedding, in which contextualization using the number of encoder and decoder layers are 2 with the dropout value 0.3. We trained our model with a batch size of 100, with the aid of Adam optimizer at 0.0005 learning rate. The learning rate has warmup update by 4000 to label smoothed cross-entropy loss function with label-smoothing value 0.1.

5 Results and Analysis

The BBPE based Transformer model was evaluated on the blind test set at five different metrics provided by the task organizer, namely BLEU (Papineni et al., 2002), BLEU-cased, TER (Snover et al., 2006), BEER2.0 (Stanojević and Sima'an, 2014), and CharacTER (Wang et al., 2016).

The obtained metrics score for each pair to each experiment is specified in Table 1. The prediction of the test set was generated by performing the best validation checkpoint. However, while comparing the BLEU score of the valid set with the test set, we obtained a difference of +3.21 for HSB→GER and +0.15 for GER→HSB pairs.

Before submitting the predictions of the test set, the BLEU scores of best and last checkpoints were almost equal, as shown in Table 2. Moreover, the vocabulary size plays a crucial role in data-driven approaches of MTs as well. Hence, we have increased the vocabulary size from 2048 to 4096 for generating BBPE, which led to a small decrement in the BLEU score. One possible reason for such decrement is the small vocabulary size that generates generalized BBPE for low-resource language.

6 Conclusion and future work

We have report the results for a Transformer-based MT system for the pair of HSB↔GER in very low resource settings. The introduced MT system works on Byte-level Byte Pair Encoding (BBPE), which yields 48.4 and 46.5 on HSB→GER and

[2] https://github.com/pytorch/fairseq

Pair	BLEU	BLEU cased	TER	BEER2.0	CharacTER
HSB-GER	48.4	47.9	0.383	0.706	0.335
GER-HSB	46.5	45.9	0.389	0.696	0.323

Table 1: Obtained scores of different metrics on the test set, provided by the task organizers

Vocab	Pair	Valid		Test
		Checkpoint (last)	Checkpoint (best)	Checkpoint (best)
2048	HSB-GER	45.92	45.19	48.4
	GER-HSB	46.62	46.35	46.5
4096	HSB-GER	45.77	45.09	-
	GER-HSB	46.96	46.24	-

Table 2: Effect on BLEU by increasing vocabulary size

GER→HSB, respectively, as the BLEU score on the test set at the vocabulary size of 2048. When the vocabulary size was increased from 2048 to 4096, lower performance was obtained on the system on either side of the pair on the validation set.

Acknowledgments

The support and the resources provided by PARAM Shivay Facility under the National Supercomputing Mission, Government of India at the Indian Institute of Technology, Varanasi are gratefully acknowledged.

References

Benyamin Ahmadnia, Javier Serrano, and Gholamreza Haffari. 2017. Persian-Spanish Low-Resource Statistical Machine Translation Through English as Pivot Language. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 24–30.

Alham Fikri Aji. 2020. In Neural Machine Translation, What Does Transfer Learning Transfer? In *Proceedings of the 2020 Annual Conference of the Association for Computational Linguistics*. Association for Computational Linguistics.

Bogdan Babych, Anthony Hartley, and Serge Sharoff. 2007. Translating from under-resourced languages: Comparing direct transfer against pivot translation. *Proceedings of the MT Summit XI*, pages 412–418.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *ICLR 2015*.

Loïc Barrault, Ondřej Bojar, Marta R Costa-Jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, et al. 2019. Findings of the 2019 Conference on Machine Translation (WMT19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 1–61.

Junyoung Chung, Kyunghyun Cho, and Yoshua Bengio. 2016. A Character-level Decoder without Explicit Segmentation for Neural Machine Translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1693–1703.

Raj Dabre, Chenhui Chu, and Anoop Kunchukuttan. 2020. A Survey of Multilingual Neural Machine Translation. *ACM Comput. Surv.*, 53(5).

Nicole Dolowy-Rybinska. 2011. A model minority. *Insight Academia*.

Nicole Dołowy-Rybińska. 2018. Learning Upper Sorbian. The problems with minority language education for non-native pupils in the Upper Sorbian grammar school in Bautzen/Budyšin. *International Journal of Bilingual Education and Bilingualism*, pages 1–15.

Ludwig Elle. 2010. Sorben–demographische und statistische Aspekte. *Vogt, Matthias Theodor, Neyer, Jürgen, Bingen, Dieter et Jan Sokol (éds.), Minderheiten als Mehrwert, Peter Lang GmbH, Frankfurt am Main*, pages 309–318.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1126–1135.

Jiatao Gu, Hany Hassan, Jacob Devlin, and Victor OK Li. 2018a. Universal Neural Machine Translation for Extremely Low Resource Languages. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 344–354.

Jiatao Gu, Yong Wang, Yun Chen, Victor OK Li, and Kyunghyun Cho. 2018b. Meta-Learning for Low-Resource Neural Machine Translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3622–3631.

Ann Irvine and Chris Callison-Burch. 2013. Combining Bilingual and Comparable Corpora for Low Resource Machine Translation. In *Proceedings of the eighth workshop on statistical machine translation*, pages 262–270.

Yunsu Kim, Petre Petrov, Pavel Petrushkov, Shahram Khadivi, and Hermann Ney. 2019. Pivot-based Transfer Learning for Neural Machine Translation between Non-English Languages. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 865–875.

Rumeng Li, Xun Wang, and Hong Yu. 2020. MetaMT, a Meta Learning Method Leveraging Multiple Domain Data for Low Resource Machine Translation. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8245–8252. AAAI Press.

Minh-Thang Luong and Christopher D Manning. 2016. Achieving Open Vocabulary Neural Machine Translation with Hybrid Word-Character Models. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1054–1063.

Preslav Nakov and Hwee Tou Ng. 2012. Improving Statistical Machine Translation for a Resource-Poor Language Using Related Resource-Rich Languages. *Journal of Artificial Intelligence Research*, 44:179–222.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A Fast, Extensible Toolkit for Sequence Modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Michael Paul, Andrew Finch, and Eiichrio Sumita. 2013. How to Choose the Best Pivot Language for Automatic Translation of Low-Resource Languages. *ACM Transactions on Asian Language Information Processing (TALIP)*, 12(4):1–17.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725.

Aditya Siddhant, Ankur Bapna, Yuan Cao, Orhan Firat, Mia Chen, Sneha Kudugunta, Naveen Arivazhagan, and Yonghui Wu. 2020. Leveraging Monolingual Data with Self-Supervision for Multilingual Neural Machine Translation. In *Proceedings of the 2020 Annual Conference of the Association for Computational Linguistics*, pages 2827–2835. Association for Computational Linguistics.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A Study of Translation Edit Rate with Targeted Human Annotation. In *Proceedings of association for machine translation in the Americas*, volume 200. Cambridge, MA.

Miloš Stanojević and Khalil Sima'an. 2014. BEER: BEtter Evaluation as Ranking. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 414–419.

Haipeng Sun, Rui Wang, Kehai Chen, Masao Utiyama, Eiichiro Sumita, and Tiejun Zhao. 2020. Knowledge Distillation for Multilingual Unsupervised Neural Machine Translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3525–3535. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. In *Advances in neural information processing systems*, pages 5998–6008.

Changhan Wang, Kyunghyun Cho, and Jiatao Gu. 2020. Neural Machine Translation with Byte-Level Subwords. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 9154–9160. AAAI Press.

Weiyue Wang, Jan-Thorsten Peter, Hendrik Rosendahl, and Hermann Ney. 2016. CharacTer: Translation Edit Rate on Character Level. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 505–510.

Zaixiang Zheng, Hao Zhou, Shujian Huang, Lei Li, Xin-Yu Dai, and Jiajun Chen. 2019. Mirror-Generative Neural Machine Translation. In *International Conference on Learning Representations*.

Low-Resource Translation as Language Modeling

Tucker Berckmann and Berkan Hiziroglu
Department of Computer Science
Brown University
tucker_berckmann@brown.edu
berkan_hiziroglu@alumni.brown.edu

Abstract

We present our submission to the very low resource supervised machine translation task at the Fifth Conference on Machine Translation. The goal of this task is to create a system which translates between German and the low-resource language Upper Sorbian. We use a decoder-only transformer architecture and formulate the translation task as language modeling. To address the low-resource aspect of the problem, we pretrain over a similar language parallel corpus. Then, we employ an intermediate back-translation step before fine-tuning. Finally, we present an analysis of the system's performance.

1 Introduction

This work describes our system for translating in both directions between German (DE) and the low-resource language Upper Sorbian (HSB). German is a widespread language with tens of millions of speakers; Upper Sorbian is a West Slavic language spoken in Germany, and it is recognized as an endangered language by UNESCO (Moseley, 2010).

This system constitutes our submission to the shared task on very-low-resource supervised machine translation at WMT20.[1] The ultimate goal of the task is to translate a blind test set from Upper Sorbian into German and *vice versa*. The task is constrained, meaning that all data sets used for training are selected from a set of corpora provided by the organizers.

Our primary contribution is our application of a decoder-only language-modeling architecture to a low-resource translation task, which to our knowledge is not well-investigated.

In Sections 2 and 3, we discuss related work and our system itself. Sections 4, 5, and 6 describe our architecture. Sections 7 and 8 contain our results and analysis.

[1] http://www.statmt.org/wmt20

2 Related Work

Current approaches to machine translation include neural networks based on encoder-decoder transformers (Vaswani et al., 2017) and sequence-to-sequence models using recurrent networks (Chen et al., 2018). In both of these methods, the system learns how to produce an intermediate representation of a text sequence as a basis for the output translation. Language-neutral representations have been explored more deeply in the context of mBert (Libovický et al., 2019).

In the case of low-resource languages, where there is an absence of adequately sized parallel corpora, recent techniques focus on transfer learning (Zoph et al., 2016), relying on monolingual corpora (Lample et al., 2018), enriching the input to the system (Irvine and Callison-Burch, 2013), or expanding it through back-translation (Sennrich et al., 2016).

Techniques related to back-translation include pseudo-labeling and self-labeling. Pseudo-labeling uses partially accurate data for training (Ratner et al., 2017) generated from knowledge bases, heuristic functions and crowdsourcing. Self-labeling is an area that lies between self-supervised learning and pseudo-labeling (Caron et al., 2018; Asano et al., 2020). The model is used to predict labels for an unlabeled dataset and then is trained on this dataset.

3 Overview

Our system uses a transformer architecture, though instead of the traditional encoder-decoder layout, we use a single decoder-only transformer as do Radford et al. (2018), formulating the translation task as a language modeling task. This architecture was suggested by Radford et al. (2019) and explored concretely by Guo et al. (2019) for widely-used languages. Unlike previous approaches, in this method

Proceedings of the 5th Conference on Machine Translation (WMT), pages 1079–1083

there is no intermediate representation of the input; instead, the translation is predicted directly through the attention mechanism.

Furthermore, in our submission, we rely on a similar-language pretraining task with a shared vocabulary, using Czech (CS) / English (EN) sentence pairs, similarly to Kocmi and Bojar (2018) and Nguyen and Chiang (2017).

Finally, we supplement these techniques with traditional back-translation, using monolingual corpora in the target languages.

4 Data Preprocessing

4.1 Datasets

In this work, we only use datasets that were made available by WMT20:

- HSB/DE parallel corpus (60K pairs)
- Monolingual HSB data (600K sentences)
- Monolingual DE news data (600K sentence subset)
- CS/EN parallel news corpus (60M pairs)

For the initial pretraining, we use CS/EN parallel data. The unlabeled and labeled HSB/DE parallel data are used for the back-translation and fine-tuning steps.

In Section 8, we also show a comparison to a reference pretraining dataset: a CS/DE parallel corpus (1.6M pairs).

4.2 Preprocessing Method

Figure 1 shows the preprocessing of the training corpus. This method of preprocessing the corpus allows us to use a single decoder-only transformer and train it on a classical language modeling task.

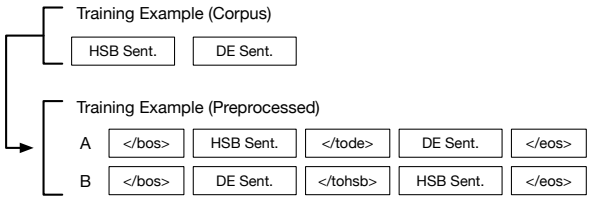


Figure 1: Preprocessing of the training corpus. The source and translation texts are concatenated with translation direction and beginning- and end-of-sequence tokens.

Since we use a CS/EN corpus on the initial pretraining step and HSB/DE corpora on the remaining steps, we create a joint byte-pair encoding which is generated by combining all of the corpora.

5 Training Method

Figure 2 shows the training method. In total, our method consists of five individual steps. These include: a pretraining step, an intermediate step made up of three sub-steps (a pre-fine-tuning step, back-translation, back-translated training), and a final fine-tuning step. The following subsections describe these steps in detail.

All of these steps (except the back-translation step, which is performed in inference mode) are performed as translation tasks using parallel corpora. The parallel corpora are either real or synthetic (in the case of the corpora resulting from back-translation).

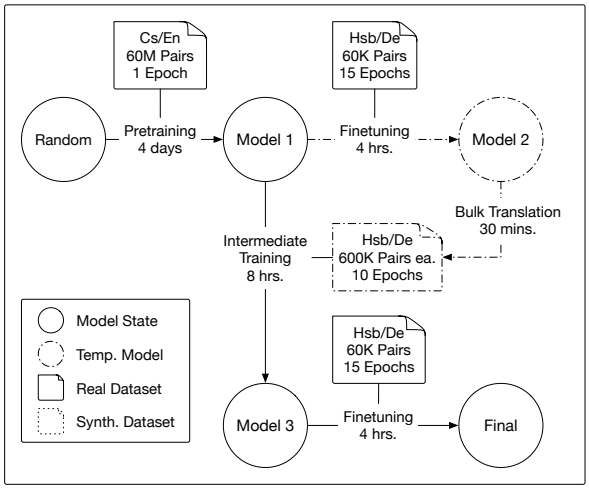


Figure 2: Training method: details are in Section 5. The dataset includes: CS/EN parallel (60M), HSB/DE parallel (60K), HSB/DE monolingal (600K each).

5.1 Initial Pretraining and Back-translation

We start by pretraining the model on a language translation task using a large (60M pair) parallel corpus consisting of Czech and English. As described by Kocmi and Bojar (2018), large (10M pair or above) parallel pretraining corpora provide significant performance gains. This is reinforced by our findings in Section 8.3.

Also, Czech and English are related to the the target languages, which can provide an additional performance benefit, according to Nguyen and Chiang (2017).

Figure 2 shows the pretraining on a CS/EN translation dataset and the first round of fine-tuning on the labeled data for the HSB/DE translation tasks.

In our method, we use the notion of *Model states*. After the pretraining step, the model reaches the *Model 1* state as depicted in Figure 2. At this step,

the model is fine-tuned and reaches the state *Model 2*. This state is used to back-translate the monolingual HSB/DE data into parallel corpora.

5.2 Back-Translated Training and Final Fine-Tuning

Output of the bulk translation is then saved and the *Model 2* state is discarded. The bulk translation is used as a parallel corpus for the back-translated training step beginning at the *Model 1* state: it consists of 600K pairs (per target language), whereby one sentence in the pair is from the monolingual corpus, and the other, parallel sentence is from the bulk translation output.

After the back-translated training, the model enters the *Model 3* state: this is the final state before the last fine-tuning. The last step is training the model in a supervised fashion on the labeled dataset.

One should note that, at this point, the model has not seen the labeled dataset yet. The state *Model 2* was trained on the labeled dataset but it is only used for the back-translation and discarded later. The final step trains the model using the highest-quality dataset: human-generated translations from the source language to the target language.

6 Implementation

We follow the GPT2 paper (Radford et al., 2019) for the model architecture, excepting hyperparameters. An overview of the system hyperparameters is shown in Table 1. We use layer normalization, a

Table 1: System Hyperparmeters

Hyperparameter	Value
Layers	4
Embedding Size	768
Attention Heads	12

standard dropout rate of 10%, and a learning rate of 5e-5 for all tasks. For the fine-tuning task, we employ L2 regularization. We train separate models for each translation direction. The total size of the model is 40M parameters.

Our choice of hyperparameters is based on a modified grid search over the attention heads, learning rate, layer count, and embedding size.

We used a single Nvidia GTX 1080TI GPU during training, and training times are shown in Figure 2. We argue that our method is time and resource efficient, easy to reproduce, and powerful.

7 Evaluation

In this section, we provide details about our implementation and the final results of the submitted system on the shared task: translation between German and Upper Sorbian.

7.1 Inference Versus Training

During the training tasks, we combine the source text, the target text, and control tokens. To use the resulting model to perform a translation of unfamiliar text, we use a slightly modified preprocessing step: we concatenate only the source text with a translation token. Since the model is trained to perform a classical language modeling task, it begins predicting the next token probabilities of the target text. We then apply beam search (with a beam width of five) to these tokens to arrive at the final translation.

7.2 Results

Table 2 shows the official BLEU score of our method on the blind test submission to WMT20. Submissions to the shared task ranged from 38.5 to 61.1 BLEU for DE to HSB translations and from 40.5 to 60.0 for HSB to DE translations.

In this section, all BLEU scores other than the blind test are calculated on the HSB/DE public test set and reference translations provided by WMT20.

Table 2: Our submission's results on the final blind test

Direction	BLEU
HSB-DE	46.0
DE-HSB	46.7

Table 3 shows a sample translation. The model's word choice is a slight generalization of the German reference, with correct grammar, spelling, and capitalization.

8 Analysis

8.1 Performance Breakdown

In order to understand the contribution of each training step to the final result, we performed experiments on different training sequences using the public HSB/DE test set provided by WMT20. The results of these experiments are reflected in Table 4. In the table, each step is cumulative and includes the steps above it: e.g., the "back-translation" step includes both fine-tuning and back-translation, but not pretraining.

Table 3: Sample Translation

Upper Sorbian Input
Otto Friedrich Bollnow mjenuje je tohodla tež hospodarske počinki.
Model output
Otto Friedrich Bollnow nennt sie deshalb auch wirtschaftliche Tugenden. **English** Therefore, Otto Friedrich Bollnow also names them economic virtues.
German Reference
Otto Friedrich Bollnow bezeichnet sie daher auch als wirtschaftliche Tugenden. **English** Therefore, Otto Friedrich Bollnow also describes them as economic virtues.

Table 4: Breakdown of BLEU score as training tasks are added: the results are cumulative

Step	DE-HSB	HSB-DE
Fine-tuning only	27.4	27.3
Back-translation	38.2	38.1
Pretraining	44.6	42.9
Blind test	46.7	46.0

We conclude from these experiments that the most significant gains came from back-translation (around 10 BLEU), followed by the pretraining step (4-6 BLEU).

For reference, we reiterate the blind test results in Table 2. The blind test results are different from the pretraining step due to differences in the data set.

8.2 Pretraining Task Selection

We considered using unsupervised learning as a pretraining task; however, a comparison of unsupervised pretraining in the target language with translation-task pretraining using related languages showed that the translation task had a greater impact on the final model's performance.

In this experiment, we compared the effect of different pretraining tasks on the model's translation performance. Recall that our architecture formulates the translation task as a language modeling task. Since the architecture acts as a language model, it is also possible to pretrain the model, without modification, on unsupervised text in the target languages.

To compare the unsupervised language modeling pretraining task with a translation pretraining task, we pretrained one model with the full HSB/DE unsupervised data set (600K sentences each, 10 epochs), a second model with a CS/DE parallel corpus (1.6M pairs, 3 epochs), and a third model with a subset of the CS/EN parallel corpus (60M pairs, 0.17 epochs), and then fine-tuned each of them using the supervised data set.

We compare these models to a baseline (fine-tuned only) model in Table 5. From these results, we conclude that the similar language translation tasks are more effective pretraining tasks than unsupervised language modeling in this context. The two related-language pretraining tasks were comparable in performance, though we only used a fraction of the CS/EN corpus due to its much larger size.

Table 5: Target-task BLEU score after fine-tuning, given pretraining tasks in various languages

Pretraining Task	Type	DE -HSB	HSB -DE
None	-	27.4	27.3
Unsupervised	HSB/DE	28.1	29.5
Translation	CS/DE	31.7	31.4
Translation	CS/EN	31.5	32.7

8.3 Pretraining Corpus Size

Finally, we examined the effect of the number of pretraining epochs on the final BLEU score. As shown in Table 6, roughly doubling the corpus size led to an increase of nearly 1.0 BLEU in the final model performance. This represents close to 20% of the performance increase we attribute to our pretraining task, which suggests that an even larger corpus, or additional pretraining epochs, would contribute further to model performance.

Table 6: Effect of 60M-pair pretraining corpus size (in epochs) on final HSB->DE BLEU score

Epochs	BLEU Score
0.42	41.4
1.00	42.3

9 Conclusion

Since our model produces high-quality translations, we have shown that a small decoder-only transformer, configured to perform classical language modeling, is an effective translation system for low-resource language pairs. Furthermore, we have shown that a similar language translation pretraining task can contribute substantially to the quality of such translation systems. Finally, we have provided an analysis of the model's components and their relative contribution to its ultimate performance.

Further investigation would be needed to understand our model's relationship to other architectures under the same data sets and pretraining tasks.

Acknowledgments

We would like to thank Prof. Eugene Charniak for the helpful discussion. We would also like to thank Prof. Ellie Pavlick for her help in reviewing the paper and the method.

References

Yuki M. Asano, Christian Rupprecht, and Andrea Vedaldi. 2020. Self-labelling via simultaneous clustering and representation learning. In *International Conference on Learning Representations*.

Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. 2018. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 132–149.

Mia Xu Chen, Orhan Firat, Ankur Bapna, Melvin Johnson, Wolfgang Macherey, George Foster, Llion Jones, Niki Parmar, Michael Schuster, Zhi-Feng Chen, Yonghui Wu, and Macduff Hughes. 2018. The best of both worlds: Combining recent advances in neural machine translation. In *ACL*.

Yinpeng Guo, Yi Liao, Xin Jiang, Qing Zhang, Yibo Zhang, and Qun Liu. 2019. Zero-shot paraphrase generation with multilingual language models. *arXiv preprint arXiv:1911.03597*.

Ann Irvine and Chris Callison-Burch. 2013. Combining bilingual and comparable corpora for low resource machine translation. In *Proceedings of the eighth workshop on statistical machine translation*, pages 262–270.

Tom Kocmi and Ondřej Bojar. 2018. Trivial transfer learning for low-resource neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 244–252, Brussels, Belgium. Association for Computational Linguistics.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Unsupervised machine translation using monolingual corpora only. In *International Conference on Learning Representations*.

Jindřich Libovickỳ, Rudolf Rosa, and Alexander Fraser. 2019. How language-neutral is multilingual bert? *arXiv preprint arXiv:1911.03310*.

Christopher Moseley, editor. 2010. *Atlas of the World's Languages in Danger*. Memory of peoples Series. UNESCO Publishing.

Toan Q. Nguyen and David Chiang. 2017. Transfer learning across low-resource, related languages for neural machine translation. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 296–301, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9.

Alexander Ratner, Stephen H Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. 2017. Snorkel: Rapid training data creation with weak supervision. In *Proceedings of the VLDB Endowment. International Conference on Very Large Data Bases*, volume 11, page 269. NIH Public Access.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer learning for low-resource neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1568–1575, Austin, Texas. Association for Computational Linguistics.

The LMU Munich System for the WMT 2020 Unsupervised Machine Translation Shared Task

Alexandra Chronopoulou, Dario Stojanovski, Viktor Hangya, Alexander Fraser

Center for Information and Language Processing, LMU Munich, Germany
{achron, stojanovski, hangyav, fraser}@cis.lmu.de

Abstract

This paper describes the submission of LMU Munich to the WMT 2020 unsupervised shared task, in two language directions, German↔Upper Sorbian. Our core unsupervised neural machine translation (UNMT) system follows the strategy of Chronopoulou et al. (2020), using a monolingual pretrained language generation model (on German) and fine-tuning it on both German and Upper Sorbian, before initializing a UNMT model, which is trained with online backtranslation. Pseudo-parallel data obtained from an unsupervised statistical machine translation (USMT) system is used to fine-tune the UNMT model. We also apply BPE-Dropout to the low-resource (Upper Sorbian) data to obtain a more robust system. We additionally experiment with residual adapters and find them useful in the Upper Sorbian→German direction. We explore sampling during backtranslation and curriculum learning to use SMT translations in a more principled way. Finally, we ensemble our best-performing systems and reach a BLEU score of 32.4 on German→Upper Sorbian and 35.2 on Upper Sorbian→German.

1 Introduction

Neural machine translation achieves remarkable results (Bahdanau et al., 2015; Vaswani et al., 2017) when large parallel training corpora are available. However, such corpora are only available for a limited number of languages. UNMT addresses this issue by using monolingual data only (Artetxe et al., 2018c; Lample et al., 2018). The performance of UNMT models is further improved using transfer learning from a pretrained cross-lingual model (Lample and Conneau, 2019; Song et al., 2019). However, pretraining also demands large monolingual corpora for both languages. Without abundant data, UNMT methods are often ineffective (Guzmán et al., 2019). Therefore, effectively translating between a high-resource and a low-resource language, in terms of monolingual data, which is the target of this year's unsupervised shared task, is challenging.

We participate in the WMT 2020 unsupervised machine translation shared task. The task includes two directions: German→Upper Sorbian (De→Hsb) and Upper Sorbian→German (Hsb→De). Our systems are constrained, using only the provided Hsb monolingual data and De NewsCrawl monolingual data released for WMT. We pretrain a monolingual encoder-decoder model on a language generation task with the Masked Sequence to Sequence model (MASS) (Song et al., 2019) and fine-tune it on both languages of interest, following Chronopoulou et al. (2020). We then train it on UNMT, using online backtranslation. We use our USMT system to backtranslate monolingual data in both languages. This pseudo-parallel corpus serves to fine-tune our UNMT model. Iterative offline backtranslation is later leveraged, yielding a performance boost. We use BPE-Dropout (Provilkov et al., 2020) as a data augmentation technique, sampling instead of greedy decoding in online backtranslation, and curriculum learning to best include the SMT pseudo-parallel data. We also use residual adapters (Houlsby et al., 2019) to translate to the low-resource language (Hsb).

Results Summary. The ensemble of our best-performing systems yields the best performance in terms of BLEU[1] among the participants of the unsupervised machine translation shared task. We release the code and our best models[2] in order to facilitate reproduction of our work and experimentation in this field. We note that we have built upon

[1] http://matrix.statmt.org/matrix/systems_list/1920

[2] https://github.com/alexandra-chron/umt-lmu-wmt2020

Proceedings of the 5th Conference on Machine Translation (WMT), pages 1084–1091
Online, November 19–20, 2020.

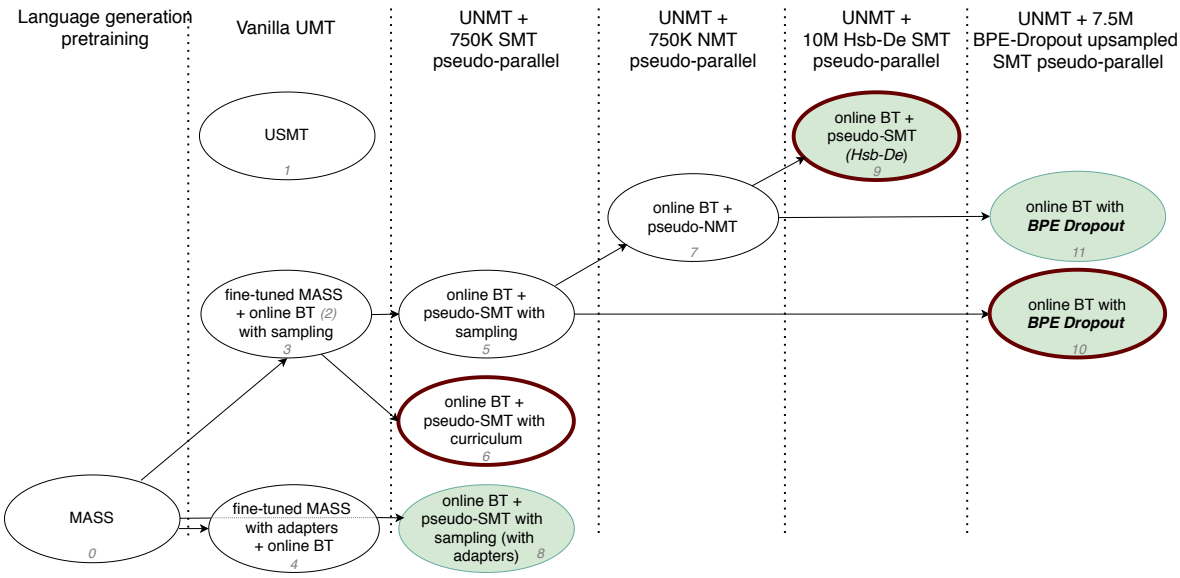


Figure 1: Illustration of our system. We denote with green the systems that were ensembled for the De→Hsb direction and with maroon the systems that were ensembled for the Hsb→De direction. Right arrows indicate transfer of weights. The numbers in gray correspond to the rows of Table 1. `Online BT` refers to the backtranslation of sentences with the actual model and updating it with the generated pseudo-parallel data. `Pseudo-SMT` refers to data obtained by backtranslating using the USMT baseline system while pseudo-NMT to our translations using system 5. The components of our approach are explained in Section 2.

the MASS codebase[3] for our experiments.

2 Model Description

Figure 1 presents all the different components of our system and how they are connected to each other. We train both an unsupervised SMT (#1) and NMT (#2) model. The UNMT model is based on a pretrained MASS model (#0), which is *monolingual* (`De`). The model is later fine-tuned on both `Hsb` and `De`. We additionally explore fine-tuning only on `Hsb` using adapters. These models are used to initialize an NMT model (#2, #4) which is trained with online backtranslation. We additionally experiment with sampling (#3) during backtranslation. The USMT model is used to backtranslate `Hsb` and `De` data. This synthetic bi-text is used to fine-tune the baseline UNMT model (#5). We use the synthetic bi-text also to fine-tune directly the adapter-augmented MASS model, while employing online backtranslation and sampling (#8). We experiment with curriculum learning (#6) to estimate the optimal way to feed the model this pseudo-parallel data. We also use our UNMT model to generate backtranslations and fine-tune existing models (#7). Further USMT-backtranslated data is used in #9. Finally, some models are fine-tuned with monolingual data which is oversampled and segmented with BPE-Dropout (#10, #11). The details of these components are outlined in the following.

2.1 Unsupervised SMT

First we describe the USMT system which we use to generate pseudo-parallel data to fine-tune our NMT system. We use *monoses* (Artetxe et al., 2018b), which builds unsupervised bilingual word embeddings (BWEs) and integrates them to Moses (Koehn et al., 2006), but apply some modifications to it.

As a first step, we build unsupervised BWEs with *fastText* (Bojanowski et al., 2017) and *VecMap* (Artetxe et al., 2018a) containing representations of 1-, 2- and 3-grams. Since the size of the available monolingual `Hsb` data is low, mapping monolingual embeddings to BWEs without any bilingual signal fails, i.e., we find no meaningful translations by manually investigating the most similar cross-lingual pairs of a few words. Instead, we rely on identical words occurring in both `De` and `Hsb` corpora as the initial seed dictionary. The BWEs are then converted to phrase-tables using cosine similarity of words and a language model is trained on the available monolingual data. The shared task organizers released a validation set which we use to tune the parameters of the system with MERT, instead of running unsupervised tuning as described in Artetxe et al. (2018b). Finally, we run 4 itera-

[3] https://github.com/microsoft/MASS

tive refinement steps to further improve the system. Other than the above, all steps and parameters are unchanged.

We use this system in inference mode to back-translate 7M De and 750K Hsb sentences. We refer to this pseudo-parallel dataset as 7.7M SMT pseudo-parallel. We also backtranslate 10M more De sentences. This dataset is later used to fine-tune one of our systems. We refer to it as 10M Hsb-De SMT pseudo-parallel.

2.2 MASS

We initialize our UNMT systems with an encoder-decoder Transformer (Vaswani et al., 2017), which is pretrained using the MASS (Song et al., 2019) objective. The model is pretrained by trying to reconstruct a sentence fragment given the remaining part of the sentence. The encoder takes a randomly masked fragment as input, while the decoder tries to predict the masked fragment. MASS is inspired by BERT (Devlin et al., 2019), but is more suitable for machine translation, as it pretrains the encoder-decoder and the attention mechanism, whereas BERT is an encoder Transformer. In order to pretrain the model, instead of training MASS on both De and Hsb, we initially train it on De. After this, we fine-tune it on both De and Hsb, following RE-LM (Chronopoulou et al., 2020). The intuition behind this is that, if we simultaneously train a cross-lingual model on unbalanced data, where X is much larger than Y, the model starts to overfit the low-resource side Y before being trained on all the high-resource language data (X). This results in poor translations. We refer to our pretrained model as FINE-TUNED MASS.

2.2.1 Vocabulary Extension for NMT

To fine-tune the pretrained De MASS model on Hsb, we need to overcome the following issue: the pretrained model uses BPE segmentation and vocabulary based only on De. To this end, we again follow RE-LM. We denote these BPE tokens as BPE_{De} and the resulting vocabulary as V_{De}. We aim to fine-tune the monolingual MASS model to Hsb. Splitting Hsb with BPE_{De} would result in heavy segmentation of Hsb words. To prevent this from happening, we learn BPEs on the joint De and Hsb corpus (BPE_{joint}). We then use BPE_{joint} tokens to split the Hsb data, resulting in a vocabulary V_{Hsb}. This method increases the number of shared tokens and enables cross-lingual transfer of the pretrained model. The final vocabulary is the union of the V_{De} and V_{Hsb} vocabularies. We extend the input and output embedding layer to account for the new vocabulary items. The new parameters are then learned during fine-tuning.

2.3 Adapters

Besides initializing our UNMT systems with FINE-TUNED MASS, we also experiment with pretraining MASS on De and fine-tuning *only* on Hsb. During fine-tuning, we freeze the encoder and decoder Transformer layers and add adapters (Houlsby et al., 2019) to each of the Transformer layers. Adapters can prevent catastrophic forgetting (Goodfellow et al., 2013) and show promising results in various tasks (Bapna and Firat, 2019; Artetxe et al., 2020). We fine-tune only the output layer, the embeddings and the decoder's attention to the encoder as well as the lightweight adapter layers.

We investigate adapters as fine-tuning in this way is considerably more computationally efficient. We also experimented with freezing the decoder's attention to the encoder as well as adding an adapter on top of it, but these architecture designs are worse in terms of perplexity during MASS fine-tuning as well as BLEU scores during UNMT.

We use the fine-tuned model to initialize an encoder-decoder Transformer, augmented with adapters. The adapter-augmented model is then trained in an unsupervised way, using online back-translation. All layers are trainable during unsupervised NMT training. We refer to this model as FINE-TUNED MASS + ADAPTERS.

2.4 Unsupervised NMT (online backtranslation)

We initialize our UNMT models with FINE-TUNED MASS. Following Song et al. (2019), we train the systems in an unsupervised manner, using online backtranslation (Sennrich et al., 2016a) of the monolingual Hsb and De data, that were also used for pretraining. As proposed in Song et al. (2019), we do not use denoising auto-encoding (Vincent et al., 2008). We use online backtranslation to generate pseudo bilingual data for training. We refer to the resulting model as UNMT BASELINE.

2.5 Sampling

We experiment with sampling instead of greedy decoding during online backtranslation. Edunov et al. (2018) show that sampling is beneficial for backtranslation compared to greedy decoding or beam search for systems trained on larger amounts

of parallel data. Although we do not use any parallel data, we assumed that our initial UNMT baseline is of reasonable quality and that sampling would be beneficial. However, in order to provide a balance, we randomly use either greedy decoding or sampling during training. The frequency with which sampling is used is a hyperparameter which we set to 0.5. Sampling temperature is set to 0.95.

2.6 Curriculum learning

Considering the high improvements achieved by including SMT backtranslated data, we conduct experiments to determine a more meaningful way to feed the data to the model using curriculum learning (Kocmi and Bojar, 2017; Platanios et al., 2019; Zhang et al., 2019). We learn the curriculum using Bayesian Optimization (BO) for which we use an open source implementation[4]. Similar work has been proposed for transfer learning (Ruder and Plank, 2017) and NMT (Wang et al., 2020). As we already have a reasonably trained NMT model, we use it to compute instance-level features for learning the curriculum. Each sentence pair from the SMT backtranslated data is represented with two features: the model scores for this pair in the *original* (backtranslation → monolingual sentence) and *reverse* direction (monolingual → backtranslation).

The weights that determine the importance of these features are learned separately for De→Hsb and Hsb→De, so that we have 4 features in total. BO runs for 30 trials. The feature weights are constrained in the range $[-1, 1]$. Each trial runs 5.4K NMT updates. The curriculum optimizes the sum of Hsb→De and De→Hsb validation perplexity. For the optimization trials, we only use the SMT backtranslated data as pseudo-parallel data and do not use online backtranslation. Finally, based on the feature weights and the features for each sentence, we sort the pseudo-parallel data and fine-tune the UNMT BASELINE with SMT backtranslations and online backtranslation. It would be interesting to study if a similar approach can be used to estimate a more optimal loading of monolingual data during MASS pretraining and UNMT.

2.7 Offline Iterative Backtranslation

We also experiment with creating synthetic training data using offline backtranslation with one of our UNMT systems (#5 in Table 1). We translate 750K De sentences to Hsb and 750K Hsb sentences to De. The resulting pseudo-parallel system is denoted as `750K NMT pseudo-parallel` corpus and is used to fine-tune the same system.

2.8 BPE-Dropout

BPE segmentation is useful in machine translation, as it efficiently addresses the open vocabulary problem. This approach keeps the most frequent words intact and splits the rare ones into multiple tokens. It builds a vocabulary of subwords and a merge table, specifying which subwords have to be merged and the priority of the merges. BPE segmentation always splits a word deterministically. Introducing stochasticity to the algorithm (Provilkov et al., 2020), by simply removing a merge from the merges with a pre-defined probability p, results in significant BLEU improvements for various languages in low- and medium-resource datasets.

We use BPE-Dropout in the following way: we oversample the Hsb monolingual data by a factor of 10 and apply BPE-Dropout. In that way, we get different segmentations of the same sentences and feed this data to the model. We also oversample the `750K SMT pseudo-parallel` corpus in the same manner, but only apply BPE-Dropout to the Hsb side. These monolingual and pseudo-parallel oversampled datasets are used to fine-tune our models. These systems perform better than our other single systems.

2.9 Ensembling

For the final models, we perform ensemble decoding with the best training models obtained in our experiments. We evaluate several combinations of model ensembles. Based on BLEU scores on the test set provided during development, we decide on two separate ensembles for De→Hsb and Hsb→De for the final submission.

3 Experiments

3.1 Data Pre-processing

In line with the rules of the WMT 2020 unsupervised shared task[5], we used 327M sentences from WMT monolingual News Crawl[6] dataset for German, collected over the period of 2007 to 2019. We also used the Upper Sorbian side of the provided parallel data as well as all of the monolingual data, a total amount of 756K sentences, provided by the

[4] https://ax.dev/

[5] http://www.statmt.org/wmt20/unsup_and_very_low_res/

[6] http://data.statmt.org/news-crawl/de/

#	Methods	De→Hsb	Hsb→De
0	MASS	5.6	7.0
1	USMT	19.3	21.4
2	⓪ UNMT baseline (fine-tuned MASS)	24.4	27.1
3	② UNMT baseline + sampling	25.4	27.4
4	⓪ UNMT baseline (fine-tuned MASS with adapters)	18.8	21.7
5	③ + online BT + pseudo-SMT + sampling	29.9	31.9
6	③ + online BT + pseudo-SMT + curriculum	30.0	32.5
6*	③ + online BT + pseudo-SMT + curriculum + sampling	30.2	32.8
7	⑤ + online BT + pseudo-NMT	29.8	33.2
8	⓪ + online BT + pseudo-SMT + sampling (with adapters)	29.0	32.3
9	⑦ + online BT + pseudo-SMT (Hsb-De)	30.0	32.7
	Data oversampling with BPE-Dropout		
10	⑤ + BPE-Dropout	30.7	33.4
11	⑦ + BPE-Dropout	31.8	34.0
12	Model Ensemble (8, 9, 10, 11)	**32.4**	**35.2**
13	Model Ensemble (6, 9, 11)	31.9	34.8

Table 1: BLEU scores of UMT for De-Hsb and Hsb-De systems. The systems with the underlined results were ensembled and used in our primary submissions. #12 is our primary system submitted to the organizers in the De→Hsb direction, while #13 is our primary system submitted in the Hsb→De direction. 6* was trained after the shared task and is not used for the final submission.

organizers. We used the provided parallel data for validation/testing (2K/2K sentences). We normalized punctuation, tokenized and true-cased the data using standard scripts from the Moses toolkit (Koehn et al., 2006). We note that we tokenized Hsb data using Czech as the language of tokenization, since these two languages are very closely related and there are no tokenization rules for Hsb in Moses.

We used BPE (Sennrich et al., 2016b) segmentation for our neural system. Specifically, we learned 32K codes and computed the vocabulary using the De data. We then also learned the same amount of BPEs on the joint corpus (De, Hsb) and computed the joint vocabulary. We extended the initial vocabulary, adding to it unseen items. We used this augmented vocabulary to fine-tune the MASS model and run all the UNMT training experiments.

3.2 Data Post-processing

We fixed the quotes to be the same as in the source sentences (German-style). We also applied a recaser using Moses (Koehn et al., 2006) to convert the translations to mixed case.

3.3 Training

Unsupervised SMT. As mentioned before, we used *fastText* (Bojanowski et al., 2017) to build 300 dimensional embeddings on the available monolingual data. We build BWEs with *VecMap* (Artetxe et al., 2018a) using identical words as the seed dictionary and restricting the vocabulary to the most frequent 200K, 400K and 400K 1-, 2- and 3-grams respectively. We used *monoses* (Artetxe et al., 2018b) as the USMT pipeline but used the available validation data for parameter tuning and ran 4 iterative refinement steps.

MASS. We use a Transformer, which consists of 6-layer encoder and 6-layer decoder with 1024 embedding/hidden size, 4096 feed-forward network size and 8 attention heads. We pretrain MASS on De monolingual data, using Adam (Kingma and Ba, 2015) optimizer with inverse square root learning rate scheduling and a learning rate of 10^{-4}. We used a per-GPU batch size of 32. We trained the model for approximately 2 weeks on 8 NVIDIA GTX 1080 Ti 11 GB GPUs. The rest of the hyperparameters follows the original MASS paper. We fine-tune MASS on both De and Hsb using the same setup, but on 4 GPUs of the same type. Fine-tuning was performed for 2 days.

Unsupervised NMT. For unsupervised NMT, we further train the fine-tuned MASS using online backtranslation. We use 4 GPUs to train each one of our UNMT models. We report BLEU using SacreBLEU (Post, 2018)[7] on the provided test set.

Unsupervised NMT + Pseudo-parallel MT. We train our UNMT systems using a pseudo-parallel supervised translation loss, in addition to the online backtranslation objective. We found out that aug-

[7] BLEU+case.mixed+numrefs.1+smooth.exp+tok.13a+version.1.4.13

menting UNMT systems with pseudo-parallel data obtained by USMT leads to major improvements in translation quality, as previous work has showed (Artetxe et al., 2018b; Stojanovski et al., 2019).

4 Results

The results of our systems on the test set provided during development are presented in Table 1. Our USMT model (#1) performs competitively, but is largely outperformed by the UNMT baseline (#2). These results are interesting considering that both systems are trained using small amounts of monolingual Hsb data. We believe that the performance of the UNMT model is largely due to the MASS fine-tuning scheme which allowed us to obtain a strong pretrained model for both languages. We also observe (#3) that mixing greedy decoding and sampling during backtranslation is beneficial compared to always using greedy decoding (#2), especially for De →Hsb which improved by 1.0 BLEU. However, it is likely that sampling is useful only if the model is of reasonable quality. We note that the adapter-augmented model (#4) is worse than the UNMT baseline.

After these initial experiments, we use the USMT model (#1) to backtranslate all Hsb monolingual data and 7M De sentences. This pseudo-parallel data is leveraged to fine-tune our UNMT models alongside online backtranslation. This approach, denoted as model #5, improves the UNMT baseline (#3) by more than 5.5 BLEU for De→Hsb and 4.5 BLEU for Hsb→De. The curriculum learning approach (#6) yields a small improvement of 0.6 BLEU for Hsb→De. Unfortunately, the curriculum learning model ran without the use of sampling. We later train the model with sampling (#6*) and obtain slight improvements in both directions.

Using NMT backtranslations in an offline manner (#7) provides for a large improvement in the Hsb→De direction, obtaining 33.2 BLEU. Further training our high scoring model #7 on USMT backtranslations, depicted as model #9, degrades performance on Hsb→De. This might indicate that USMT backtranslations alone are not very important for high performance, but simply adding any kind of pseudo-parallel data during training.

The adapter-augmented model with USMT backtranslations (#8) manages to close the gap to the baseline model. Comparing #5 and #8, we can see that the model with adapters is worse by 0.9 BLEU on De→Hsb, but better by 0.4 on Hsb→De. Due to time constraints, we train #4 and #8 in parallel and #8 is not fine-tuned from #4. Overall, adapters are a promising research direction as they lead to faster MASS fine-tuning and comparable performance.

We observe considerable improvements using BPE-Dropout. As noted before, we oversample the parallel and Hsb monolingual data and apply BPE-Dropout only on Hsb. We use this data to fine-tune some of our already trained models, specifically #5 and #7 which results in models #10 and #11, respectively. This approach improves the Hsb→De direction by up to 1.5 BLEU and up to 1.0 BLEU for De→Hsb. System #11 proved to be our best single system in both translation directions. We hypothesize that using BPE-Dropout while simultaneously oversampling the data provides for a data augmentation effect. In future work, it would be interesting to decouple these two steps and measure their effect separately.

Ensembling further boosts performance. Ensemble #12 is used for De→Hsb and #13 for Hsb→De. We note that while computing ensemble BLEU scores during development, we did not fix the issue with German-style quotes. This resulted in ensemble #13 obtaining better scores on Hsb→De. We later fix the quotes issue and find out that ensemble #12 is better on both translation directions and is the best system overall.

5 Conclusion

In this paper, we present the LMU Munich system for the WMT 2020 unsupervised shared task for translation between German and Upper Sorbian. Our system is a combination of an SMT and an NMT model trained in an unsupervised way. The UNMT model is trained by fine-tuning a MASS model, according to the recently proposed RE-LM approach. The experiments show that the MASS fine-tuning technique is efficient even if little monolingual data is available for one language and results in a strong UNMT model. We also show that using pseudo-parallel data from USMT and UNMT backtranslations improves performance considerably. Furthermore, we show that oversampling the low-resource Upper Sorbian and applying BPE-Dropout, which can effectively be seen as data augmentation, results in further improvements. Adapters in MASS fine-tuning provided for a balance between performance and computational efficiency. Finally, smaller but noticeable gains are obtained from us-

ing curriculum learning and sampling during decoding in backtranslation.

Acknowledgments

This work was supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 640550) and by the German Research Foundation (DFG; grant FR 2829/4-1). We would like to thank Jindřich Libovický for fruitful discussions regarding the use of BPE-Dropout as a data augmentation technique.

References

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018a. A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 789–798.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018b. Unsupervised statistical machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 3632–3642.

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. 2018c. Unsupervised neural machine translation. In *International Conference on Learning Representations*.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. On the cross-lingual transferability of monolingual representations. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations*.

Ankur Bapna and Orhan Firat. 2019. Simple, scalable adaptation for neural machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing*, pages 1538–1548.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, pages 135–146.

Alexandra Chronopoulou, Dario Stojanovski, and Alexander Fraser. 2020. Reusing a Pretrained Language Model on Languages with Limited corpora for Unsupervised NMT. *arXiv preprint arXiv:2009.07610*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 489–500.

Ian J Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. 2013. An empirical investigation of catastrophic forgetting in gradient-based neural networks. *arXiv preprint arXiv:1312.6211*.

Francisco Guzmán, Peng-Jen Chen, Myle Ott, Juan Pino, Guillaume Lample, Philipp Koehn, Vishrav Chaudhary, and Marc'Aurelio Ranzato. 2019. The FLORES evaluation datasets for low-resource machine translation: Nepali–English and Sinhala–English. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing*, pages 6100–6113.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proceedings of the International Conference on Machine Learning*.

Diederick P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*.

Tom Kocmi and Ondřej Bojar. 2017. Curriculum learning and minibatch bucketing in neural machine translation. In *Proceedings of the International Conference Recent Advances in Natural Language Processing*, pages 379–386.

Philipp Koehn, Marcello Federico, Wade Shen, Nicola Bertoldi, Ondrej Bojar, Chris Callison-Burch, Brooke Cowan, Chris Dyer, Hieu Hoang, Richard Zens, et al. 2006. Open source toolkit for statistical machine translation: Factored translation models and confusion network decoding. In *Final Report of the 2006 JHU Summer Workshop*.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems*, page 7057–7067.

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Phrase-based & neural unsupervised machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 5039–5049.

Emmanouil Antonios Platanios, Otilia Stretcu, Graham Neubig, Barnabas Poczos, and Tom Mitchell. 2019. Competence-based curriculum learning for neural machine translation. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1162–1172.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Conference on Machine Translation: Research Papers*, pages 186–191.

Ivan Provilkov, Dmitrii Emelianenko, and Elena Voita. 2020. BPE-Dropout: Simple and effective subword regularization. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 1882–1892.

Sebastian Ruder and Barbara Plank. 2017. Learning to select data for transfer learning with Bayesian optimization. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 372–382.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 86–96.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 1715–1725.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. MASS: Masked Sequence to Sequence pre-training for language generation. In *Proceedings of the International Conference on Machine learning*, pages 5926–5936.

Dario Stojanovski, Viktor Hangya, Matthias Huck, and Alexander Fraser. 2019. The LMU Munich unsupervised machine translation system for WMT19. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 393–399.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, page 5998–6008.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. 2008. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the International Conference on Machine Learning*, pages 1096–1103.

Wei Wang, Ye Tian, Jiquan Ngiam, Yinfei Yang, Isaac Caswell, and Zarana Parekh. 2020. Learning a multi-domain curriculum for neural machine translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 7711–7723.

Xuan Zhang, Pamela Shapiro, Gaurav Kumar, Paul McNamee, Marine Carpuat, and Kevin Duh. 2019. Curriculum learning for domain adaptation in neural machine translation. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1903–1915.

UdS-DFKI@WMT20: Unsupervised MT and Very Low Resource Supervised MT for German↔Upper Sorbian

Sourav Dutta[1]*, Jesujoba O. Alabi[1,3]*, Saptarashmi Bandyopadhyay[2], Dana Ruiter[1], Josef van Genabith[1,3]
[1]Saarland University, Saarbrücken, Germany
[2]University of Maryland, College Park, MD 20742
[3]DFKI GmbH, Saarbrücken, Germany
souravd@coli.uni-saarland.de
sapta.band59@gmail.com, druiter@lsv.uni-saarland.de
{jesujoba_oluwadara.alabi, josef.van_genabith}@dfki.de

Abstract

This paper describes the UdS-DFKI submission to the shared task for unsupervised machine translation (MT) and very low-resource supervised MT between German (de) and Upper Sorbian (hsb) at the Fifth Conference of Machine Translation (WMT20). We submit systems for both the supervised and unsupervised tracks. Apart from various experimental approaches like bitext mining, model pretraining, and iterative back-translation, we employ a factored machine translation approach on a small BPE vocabulary.

1 Introduction

This paper describes the UdS-DFKI submission to the unsupervised and very low resource supervised tasks of WMT20 for German to Upper Sorbian (*de→hsb*) and Upper Sorbian to German (*hsb→de*). Our submitted systems are constrained for the very low resource supervised and unconstrained for the unsupervised task, in that we use Wikipedia dumps as additional data.

Current machine translation systems that deal with low-resource languages are based on unsupervised neural machine translation, semi-supervised methods and pre-trained models leveraging monolingual data (Guzmán et al., 2019), and multilingual systems among others. In this work, we explore different systems which include baseline NMT, factored NMT (Sennrich and Haddow, 2016a), iterative backtranslation, self-supervised NMT (SSNMT) (Ruiter et al., 2019) and pretraining with XLM (Lample and Conneau, 2019) using transformer-base models (Vaswani et al., 2017) for the training of the systems.

This paper begins by presenting the data we used for the tasks and the preprocessing pipeline (Section 2). This is followed by an overview of the training setup (Section 3) and the methods we applied (Section 4). Section 5 summarizes our findings, followed by a discussion of the results in Section 6. We conclude the paper and propose some possible future work in Section 7.

2 Data

Unsupervised Task For Sorbian, we use the given data from the Sorbian Institute (Ins_{hsb}), from Witaj Sprachzentrum (Witaj_{hsb}), and web-scraped data (Web_{hsb}). Table 1 gives a summary of the data we use in the unsupervised track. We use the Europarl (EP_mono_{de}, (Koehn, 2005)) and News Commentary (NC_mono_{de}, (Barrault et al., 2019)) datasets for the monolingual German data. Apart from this, we also use Wikipedia Dumps[1] for both German and Upper Sorbian. We extract articles using Wikiextractor[2], which are aligned using Wikipedia *langlinks*[3] to create a comparable corpus for SSNMT extraction.

Supervised Task Apart from the provided parallel data, we use high-quality EUROPARL (EP, (Koehn, 2005)) and medium-quality JW300 (Agić and Vulić, 2019; Tiedemann, 2012) corpora for *de-cs-hsb*. For parallel text mining with LASER (Schwenk, 2018; Artetxe and Schwenk, 2019), we use the combination of all the monolingual corpora of German and Upper Sorbian from the unsupervised section of Table 1, which is discussed in detail later in Section 4.3.

Preprocessing Our preprocessing steps include normalization, tokenization, deduplication, and truecasing. We attach feature labels related to the source language (<src>), target language (<tgt>), and the data quality (<quality>), for

* Equal contribution

[1] https://dumps.wikimedia.org/ (March 2020)
[2] https://github.com/attardi/wikiextractor
[3] https://www.mediawiki.org/wiki/API:Langlinks

Proceedings of the 5th Conference on Machine Translation (WMT), pages 1092–1098
Online, November 19–20, 2020.

every individual sentence. The quality of a sentence depends on the corpus it is from and the quality tags of <low>, <medium>, or <high> are added to all sentences of the corpora according to the quality labels assigned to the data provided for the shared task: e.g. Ins_{hsb} is high quality Sorbian. A typical sentence from the corpus after our preprocessing pipeline has the following format:

```
<src> <tgt> <quality> sentence
```

After factoring (4.2), we proceed to apply joint byte-pair encoding (BPE) (Sennrich et al., 2016b) on the corpora to finally get our preprocessed data which we use for training all our NMT models. Unless specified otherwise, we use a default of 5k merge operations.

Corpus	# Sentences	# Tokens
Unsupervised		
Ins_{hsb}	339k	5,044k
Witaj_{hsb}	222k	2,672k
Web_{hsb}	134k	1,677k
EP_mono_{de}	2,107k	55,557k
NC_mono_{de}	422k	8,942k
Wiki_{de}	833k	36,531k
Wiki_{hsb}	76k	2,402k
Supervised		
Bitext_{de}	60k	1,002k
Bitext_{hsb}	60k	737k
EP_{de}	568k	13,098k
EP_{cs}	568k	11,571k
JW300_{de}	1,179k	20,888k
JW300_{cs}	1,179k	19,144k
Dev & Test		
Dev20_{de}	2k	24k
Dev20_{hsb}	2k	21k
DevTest20_{de}	2k	24k
DevTest20_{hsb}	2k	22k

Table 1: Statistics (in thousands) of different corpora used for the unsupervised and supervised tasks.

3 Systems

MT Systems We train all our models using the Transformer-base architecture in the OpenNMT-py (Klein et al., 2017) framework extended for SSNMT[4] (Ruiter et al., 2019). The setting for the Transformer base is the same as in Vaswani et al. (2017) with 6 encoder-decoder layers after having explored other options of Transformer depth. We set the dropout to 0.4 in all experiments. We use *adam* (Kingma and Ba, 2014) for optimization with $\beta_1 = 0.9$ and $\beta_2 = 0.998$. The learning rate is varied from 0 to 2 with a warm update of 4000 and decayed using *noam*. Lower values of learning rate were avoided due to slower training and lower accuracy scores. We use a batch size of 50. The Phrase-Based Statistical MT systems (PBSMT) are standard *Moses* (Koehn et al., 2007) systems trained without applying BPE to the data.

Initialization The NMT models are initialised with cross-lingual word embeddings calculated on the monolingual corpora using `word2vec`[5] (Mikolov et al., 2013) (skip-gram) and unsupervised `VecMap`[6] (Artetxe et al., 2017).

Pre-trained Sentence Representations For XLM (Lample and Conneau, 2019), we pre-train and fine-tune the model using drop out of 0.1, batch size of 32 with a joint BPE of $10k$ ($10k$ showed better results for XLM), learning rate of 0.0001, and a sequence length of 265 using 512 and 1024 embedding dimensions respectively.

Evaluation Metric We use BLEU[7] (Papineni et al., 2002) scores to evaluate the performance of our trained models.

4 Techniques

4.1 Iterative Backtranslation

For the unsupervised task, we use an SSNMT system as described in Ruiter et al. (2019) to extract parallel sentences from the Wikipedia dumps. SSNMT jointly and iteratively extracts parallel data, and learns the MT task on the extracted parallel data. The resulting trained NMT model is our base model (M_0).

For iterative back-translation (Hoang et al., 2018), we take a model M_i and use it to translate the *hsb* monolingual data mono_{hsb} and EP_mono_{de} to generate mono^i_{de} and EP_mono^i_{hsb} respectively. Following Sennrich et al. (2016a), we use the generated data at iteration i on the source side with

[4] https://github.com/ruitedk6/comparableNMT

[5] https://github.com/tmikolov/word2vec

[6] https://github.com/artetxem/vecmap

[7] We use the *Moses* multi-bleu script for evaluation. https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/multi-bleu.perl

the original data on the target to train a new model M_{i+1}. This is done iteratively, in our case until $i = 5$.

As the translation quality of M_0 is very low, this model is replaced by a PBSMT system which is trained on the data that M_0 has extracted, in order to generate the back-translation to be used for M_1.

Model	BLEU	
	hsb→de	*de→hsb*
M_0	6.46	6.09
M_1	8.53	8.31
M_2	9.81	10.04
M_3	10.47	13.51
M_4	**11.31**	11.57
M_5	9.13	**13.61**

Table 2: BLEU scores of iterative-backtranslation models per iteration, calculated on Dev20.

The resulting BLEU scores on Dev20 for each of the iterations is shown in Table 2. The best performance for *hsb→de* is achieved at $i = 4$ (11.31 BLEU) and for *de→hsb* at $i = 5$ (13.61 BLEU). These constitute two of the models submitted to the unsupervised task.

4.2 Factorization

Limited monolingual language analysis tools and few linguistic analysis tools with acceptable performance are available for low-resource (LowRes) languages. In our experiments, we explore factored machine translation (García-Martínez et al., 2016; Sennrich and Haddow, 2016b; Koehn and Knowles, 2017). This approach can play a significant role in increasing grammatical coherence. Syntactic and semantic information can be useful to generalize neural models trained on parallel corpora.

We augment our parallel data to include factors like *lemma* (using Snowball Stemmer (Porter, 2001)) and *PoS* tags (using *spaCy*[8] open source library (Honnibal and Montani, 2017)) for German words. The language-agnostic UDPipe trainable pipeline (Straka et al., 2016) has been used for lemmatization and PoS tagging for Sorbian words. We follow an approach similar to Bandyopadhyay (2019, 2020), where we factor the data at word-level to include the root word (*lemma*) and the part of speech (*PoS*) of each word along with the word itself, each component separated with a pipe (|) symbol.

```
word_token | lemma | PoS
```

Byte-pair encoding is implemented after factorization. After training the model, the test dataset on the source side of the language pair is used to obtain the output dataset on the target side of the language pair. Once testing is done, the data is again decoded using the trained BPE model before.

For the **supervised** task, we submit a German to Upper Sorbian factorized model on the German side of the parallel corpus which resulted in 40.9 and 40.3 cased BLEU score.

For the **unsupervised** task, Upper Sorbian to German factorization on the best-performing SS-NMT model improves the BLEU score by 0.1 to 9.0 on Test20 in comparison to the non-factorized model.

The results of the factored models are reported in Tables 3 (supervised) and 4 (unsupervised).

BPE	de (fac.)→hsb	hsb (fac.)→de
2k	31.01	37.09
5k	**41.15**	32.17
10k	35.67	**38.23**
20k	34.70	37.62

Table 3: Supervised Source Factored NMT systems with BLEU scores on DevTest20.

System	BLEU
de→hsb (fac.)	5.67
de (fac.)→hsb (fac.)	6.03
hsb→de (fac.)	7.24
hsb (fac.)→de (fac.)	**7.49**

Table 4: Unsupervised Factored NMT systems with BLEU scores for 10k BPE on DevTest20.

4.3 Data Mining with LASER

We use LASER (Schwenk, 2018; Artetxe and Schwenk, 2019) to filter and mine parallel sentences from a list of monolingual corpora of both German and Upper Sorbian. For German, we use the Wiki_{de}, EP_mono_{de}, and NC_{de} corpora, while

[8] https://github.com/explosion/spaCy

for the Upper Sorbian counterpart, we use the monolingual corpora Ins_{hsb}, Witaj_{hsb}, Web_{hsb}, and Wikipedia dumps (Wiki_{hsb}) as mentioned in Table 1. We explore a range of LASER extraction threshold values (1.03, 1.04, 1.05, 1.06, and 1.07) for this process. Table 5 gives a summary of the number of parallel sentences extracted from the monolingual corpora combinations from both languages using different threshold values. Using a lower threshold value extracts a higher number of parallel sentences but the quality gradually deteriorates as the threshold value decreases. We train NMT models on parallel sentences from each threshold and find that 1.04 gives comparatively better results than others. We use the model M_4 from iterative backtranslation (Table 2) as the baseline and then add the extracted sentences to check if the performance improves. However, all the resulting BLEU scores using additional LASER data are much lower than those of the iterative backtranslation baseline models reported in Table 2, indicating poor quality of the LASER extractions.

Threshold	# Sentences
1.03	18,979
1.04	**9,609**
1.05	5,200
1.06	2,806
1.07	1,646

Table 5: Number of parallel sentences mined using LASER with different threshold values.

4.4 Pre-training with Cross-lingual Language Model XLM

We explore the option of using pre-trained models with different embedding sizes to improve the performance of our system in the unsupervised task. We collected the sentence pairs from Wikipedia extracted with SSNMT. Also we collected back-translations for the monolingual data provided for the task using iterative backtranslation as explained in Section 4.1. We then pre-train XLM for de-hsb using all the monolingual data except Wiki_{de} and Wiki_{hsb}. We then fine-tune the pre-trained model for the supervised translation task using the parallel data from M_0 and back-translations taken from M_4 and M_5. Table 6 shows the resulting BLEU scores for this task on Dev20 and DevTest20.

XLM Embedding Size	BLEU	
	hsb→de	*de→hsb*
Dev20		
512	8.84	**8.41**
1024	**8.91**	8.15
DevTest20		
512	**7.58**	**7.29**
1024	7.44	6.78

Table 6: BLEU scores of pre-training with XLM on Dev20 and DevTest20.

5 Results

Tables 7 (submitted systems) and 8 (unfactored baseline systems) show a summary of all BLEU scores.

Model	BLEU		
	Dev20	DevTest20	Test20
Unsupervised			
de→hsb	13.6	9.9	10.3
hsb→de	11.3	8.1	8.9
hsb (fac.)→de	9.8	8.7	9.0
Supervised			
de (fac.)→hsb	44.34	41.15	40.9

Table 7: BLEU scores for the submitted models on the Dev20, DevTest20, and Test20 datasets.

Unsupervised Parallel data extracted with self-supervised NMT on Wikipedia dumps data and iterative back-translation on mono_{hsb} EP_mono_{de} were used to train the models. For the unsupervised track, we submit three NMT models trained in the directions from unfactored German to unfactored Upper Sorbian (*de→hsb*), from unfactored Upper Sorbian to unfactored German (*hsb→de*), and from factored Upper Sorbian to unfactored German (*hsb*(fac.)*→de*). The iterative backtranslation model M_5 (Table 2) for *de→hsb* obtains a BLEU score of **9.0** on the WMT blind test data. The *hsb→de* model (M_4 in Table 2) achieves a BLEU score of **8.9** while the same model with a factored Upper Sorbian source slightly pushes the BLEU score to **9.0**.

Supervised For the supervised task, we submit a single *de*(fac.)→*hsb* NMT model (refer Table 3) where the German side is factored. The model achieves a BLEU score of **40.9** on the WMT blind test data.

6 Discussion

System	BPE	de→hsb	hsb→de
PBSMT		36.93	37.65
bilingual de-hsb	2k	**41.16**	**40.57**
	5k	37.51	37.47
	15k	37.68	36.79
	30k	36.02	35.64
multilingual de-cs-hsb	2k	28.20	30.98
	5k	34.05	36.07
	15k	32.98	36.61
	30k	29.31	36.89

Table 8: Supervised NMT systems with BLEU scores on DevTest20.

We experimented with different methods in this shared task for both the supervised as well as unsupervised tracks. The major challenge in this task was the small amount of good quality training data as Upper Sorbian is a very low resource language. Parallel sentence extraction demands the availability of good quality data. Schwenk (2018) and Artetxe and Schwenk (2019) mention that the pre-trained LASER model seems to generalize well for minor languages and dialects including Sorbian[9], but Upper Sorbian itself is not among the languages on which the model is actually trained. As a result, LASER does not seem to give very good results for Upper Sorbian. SSNMT (Ruiter et al., 2019) however was able to learn better semantic representations and extracted quality sentence pairs from Wikipedia articles.

The lack of sufficient data for training is also one of the reasons why pre-trained language models using XLM did not give satisfactory results. The second reason is the low quality of the back-translations that were used for fine-tuning.

We have used factored machine translation where we include the *lemma* and the *PoS* of each word along with it in the corpora. Due to the lack of a proper lemmatizer for Upper Sorbian, we used UDPipe (Straka et al., 2016) for Czech as it is another language from the Slavic family. However, there are obvious linguistic differences in both the languages due to which a Czech morphological tool will not work perfectly for Upper Sorbian. This is also the reason why our *de-cs-hsb* multilingual NMT systems (Table 8) did not achieve satisfactory results. NMT models with factored source sentences improved the performances of our models by a small margin.

We have observed that a smaller BPE vocabulary is generally better for low-resource languages as expected. Here we have chosen an optimal BPE vocabulary size as choosing even smaller BPE size values would result in almost character-level segmentation. We also realise that the availability of more quality data could have improved our systems as we can first pre-train language models on good quality monolingual text data using XLM and use this as the initial model for iterative backtranslation as in the SSNMT approach. We believe that this will generate better results.

7 Conclusion and Future Work

This paper describes the UdS-DFKI submission to the shared task of unsupervised and very low resource supervised machine translation between German and Upper Sorbian at WMT20. For all our systems, we have used the standard Transformer-base architecture. We have extracted parallel data from Wikipedia dumps using SSNMT (Ruiter et al., 2019), followed by iterative back-translation for the unsupervised task. For the supervised track, we have tried to factor morphological information into our data to improve our results further. For the constrained supervised task, we achieve 40.9 BLEU for *de*(fac.)→*hsb*. We obtain BLEU scores of 10.3, 8.9, and 9.0 for the *de*→*hsb*, *hsb*→*de*, and *hsb*(fac.)→*de* translation directions respectively in the unsupervised track.

As discussed in Section 6, one approach for future work is to combine XLM pre-training along with SSNMT directly to improve system initialization. It would be interesting to explore linguistic and syntactic information from other closely-related languages to further enhance the performance of the multilingual models.

Acknowledgments

The authors thank the German Research Center for Artificial Intelligence (DFKI GmbH) for pro-

[9] https://github.com/facebookresearch/LASER#supported-languages

viding the necessary infrastructure to run all the experiments.

References

Željko Agić and Ivan Vulić. 2019. Jw300: A wide-coverage parallel corpus for low-resource languages. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3204–3210.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2017. Learning bilingual word embeddings with (almost) no bilingual data. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 451–462.

Mikel Artetxe and Holger Schwenk. 2019. Margin-based parallel corpus mining with multilingual sentence embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3197–3203.

Saptarashmi Bandyopadhyay. 2019. Factored neural machine translation at loresmt 2019. In *Proceedings of the 2nd Workshop on Technologies for MT of Low Resource Languages*, pages 68–71.

Saptarashmi Bandyopadhyay. 2020. Factored neural machine translation on low resource languages in the covid-19 crisis.

Loïc Barrault, Ondřej Bojar, Marta R Costa-Jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, et al. 2019. Findings of the 2019 conference on machine translation (wmt19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 1–61.

Mercedes García-Martínez, Loïc Barrault, and Fethi Bougares. 2016. Factored neural machine translation. *CoRR*, abs/1609.04621.

Francisco Guzmán, Peng-Jen Chen, Myle Ott, Juan Pino, Guillaume Lample, Philipp Koehn, Vishrav Chaudhary, and Marc'Aurelio Ranzato. 2019. The flores evaluation datasets for low-resource machine translation: Nepali–english and sinhala–english. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6100–6113.

Vu Cong Duy Hoang, Philipp Koehn, Gholamreza Haffari, and Trevor Cohn. 2018. Iterative back-translation for neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 18–24, Melbourne, Australia. Association for Computational Linguistics.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. In *International Conference for Learning Representations (ICLR)*.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M Rush. 2017. Opennmt: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86. Citeseer.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. *CoRR*, abs/1706.03872.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Martin Porter. 2001. Snowball: A language for stemming algorithms.

Dana Ruiter, Cristina Espana-Bonet, and Josef van Genabith. 2019. Self-supervised neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1828–1834.

Holger Schwenk. 2018. Filtering and mining parallel data in a joint multilingual space. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 228–234.

Rico Sennrich and Barry Haddow. 2016a. Linguistic input features improve neural machine translation. In *Proceedings of the First Conference on Machine Translation: Volume 1, Research Papers*, pages 83–91.

Rico Sennrich and Barry Haddow. 2016b. Linguistic input features improve neural machine translation. *CoRR*, abs/1606.02892.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725.

Milan Straka, Jan Hajic, and Jana Straková. 2016. Udpipe: trainable pipeline for processing conll-u files performing tokenization, morphological analysis, pos tagging and parsing. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4290–4297.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey. European Language Resources Association (ELRA).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, pages 5998–6008.

Data Selection for Unsupervised Translation of German–Upper Sorbian

Lukas Edman **Antonio Toral** **Gertjan van Noord**
Center for Language and Cognition
University of Groningen
{j.l.edman, a.toral.ruiz, g.j.m.van.noord}@rug.nl

Abstract

This paper describes the methods behind the systems submitted by the University of Groningen for the WMT 2020 Unsupervised Machine Translation task for German–Upper Sorbian. We investigate the usefulness of data selection in the unsupervised setting. We find that we can perform data selection using a pretrained model and show that the quality of a set of sentences or documents can have a great impact on the performance of the unsupervised neural machine translation (UNMT) system trained on it. Furthermore, we show that document-level data selection should be preferred for training the state-of-the-art UNMT model, the XLM model, when possible. Finally, we show that there is a trade-off between quality and quantity of the data used to train UNMT systems.

1 Introduction

Unsupervised Neural Machine Translation (UNMT) has recently become the dominant paradigm for unsupervised MT, with the advent of cross-lingual language model pretraining as used in the XLM model (Conneau and Lample, 2019). However, much of the existing research in UNMT assumes that the amount of data available for one language is roughly equivalent to the other. The WMT 2020 Unsupervised Machine Translation task is unique in that monolingual data is abundant for one language (German), with hundreds of millions of sentences available, and sparse for the other (Upper Sorbian), which only has around 750 thousand sentences available. With a wealth of data available on the German side, it is natural to ask: how can we best use this data? Viewing this under the lens of data selection, we break this broad question down into 3 concrete sub-questions, tailored for the unsupervised setting. They are as follows:

- How can we determine the quality of training data?
- What kinds of data selection are best for training an XLM model?
- Is quality or quantity more important when it comes to training data for UNMT?

Section 2 describes the general setup pertaining to every experiment, including datasets, data processing steps, model architecture, and training details. In Section 3, we detail our individual experiments and their corresponding results. Finally, in Section 4, we make our conclusions and discuss paths for future work.

2 Setup

For Upper Sorbian, we use the 3 monolingual datasets provided by the Sorbian Institute, the Witaj Sprachzentrum, and the web data from CIS, LMU. We also use the Upper Sorbian side of the parallel corpus from `train.hsb-de.hsb.gz`. For German, we use monolingual data from News Crawl and Common Crawl. For validation and testing, we use the data provided in `devtest.tar.gz`.

All data is tokenized and truecased using the Moses toolkit (Koehn et al., 2007). For BPE segmentation (Sennrich et al., 2016), we apply a joint segmentation for both languages. This is done by first taking a sample of the German data of the same length as the Upper Sorbian data (around 750 thousand sentences). The BPE codes are learned and applied using FastBPE.[1] After BPE is applied, we remove duplicate sentences while retaining the order of the corpora.[2]

We used the XLM model (Conneau and Lample, 2019) using the default parameters, with the excep-

[1] https://github.com/glample/fastBPE

[2] For document-level filtering, we do not remove duplicates.

Proceedings of the 5th Conference on Machine Translation (WMT), pages 1099–1103

tion of allowing for sentences of max length 200 rather than 100.[3] The language model pretraining step includes only masked language modelling, and training is limited to 24 hours. The NMT step is also limited to 24 hours, with the additional stopping criterion of no improvement on the DE→HSB validation set for 10 epochs.[4]

3 Experiments

For all of our data selection experiments, we start by training an initial model. Our initial model is trained on 10 million German sentences and all of the available Upper Sorbian sentences. The 10 million German sentences include all of the data from years 2007 and 2010, and the remaining sentences are taken from 2014.[5] Our initial model achieves BLEU scores of 17.43 and 19.05 for DE→HSB and HSB→DE respectively.

3.1 Data Selection

We apply two forms of data selection: sentence-level and document-level. As we have an abundance of German data ($\mathcal{D}$) and limited Upper Sorbian data ($\mathcal{H}$), we are only concerned with data selection for German. To select from $\mathcal{D}$, we first must score our data in terms of its potential to improve the performance of our NMT model. Drawing inspiration from Moore and Lewis (2010), our scoring function is as follows:

$$Score(s) = \frac{LM_{\mathcal{H}\rightarrow\mathcal{D}'}(s) - LM_{\mathcal{D}}(s)}{|s|}$$

In this equation, s refers to any sentence in the German data, $|s|$ to its token length, $LM_{\mathcal{X}}(s)$ to the log probability of s using a language model trained on dataset $\mathcal{X}$, and $\mathcal{H} \rightarrow \mathcal{D}'$ to the dataset obtained by translating $\mathcal{H}$ into German using the initial system. A high scoring sentence is thus a sentence that has a high probability according to the Upper Sorbian language model compared to that of the German language model.[6]

[3]The max length increase was found to perform slightly better in early testing.

[4]Both steps are limited to 24 hours as there was little to no improvement observed beyond 24 hours in preliminary tests.

[5]We choose these years because we found that the frequencies of "20XX" in the Upper Sorbian data peak at 2005, 2010, and 2014, and 2007 is the earliest News Crawl data available.

[6]The intuition behind subtracting the score of the German language model is that without it a sentence may have a high score due to it containing frequent words in general (e.g. "the") rather than words that are particularly frequent in the Upper Sorbian dataset (e.g. "Sorbia").

Selection Type	DE→HSB	HSB→DE
Sentence - Low	5.21	5.91
Sentence - Random	**16.98**	**18.45**
Sentence - High	15.08	18.05
Document - Low	9.32	8.46
Document - Random	17.03	18.19
Document - High	**17.60**	**19.23**

Table 1: BLEU scores for XLM trained on data selected with the lowest and highest sentence and document-level scores, as well as randomly selected sentences and documents.

The language model we use is KenLM (Heafield et al., 2013). We use a trigram model, with all other parameters being the default values. Since we require a portion of the German dataset to train the model, we choose N sentences randomly, with N being equal to the number of sentences in $\mathcal{H}$.[7] These sentences are not included during the selection process.

For sentence-level selection, we simply order each sentence based on score and select the sentences with the highest scores. For document-level selection, we score each document by averaging its sentence-level scores, and select the documents with the highest scores.

To answer our first research question, we show that systems trained on the highest scoring sentences and documents perform significantly better than those trained on the lowest scoring sentences and documents. For this experiment, we start with 10 million sentences from News Crawl 2015, and score each sentence and document. We then train models on the 2 million lowest and highest scoring sentences, as well as the lowest and highest scoring documents which total 2 million sentences in length. The results are shown in Table 1.

The results show a drastic improvement from using the lowest quality sentences to the highest according to our scoring function. This applies both at the sentence and document level. However only document-level filtering outperforms random selection. We speculate that this is due to a potential lack of variety in the sentence-level filtering, as it may select sentences with substantial trigram overlap, due to their similarly high score. This would be less of an issue on the document-level, since there is a smaller likelihood for two documents to have a high degree of overlap. A potential solution

[7]The choice of N follows Moore and Lewis (2010).

to this lack of variety would be to select sentences sequentially, enforcing a word overlap constraint. This would limit the number of words a sentence could share with previously selected sentences.

3.2 Document-level versus sentence-level

We see from Table 1 that document-level selection outperforms sentence-level selection. This could be for 2 reasons: either the sentences selected are higher quality on average or the language model pretraining step for the XLM model benefits more from documents than sentences. To further explain the latter reason, the pretraining step for XLM uses streams of text which can contain multiple sentences, so sentences being in order should be beneficial for training the language model. To test this, we take the document-level selected sentences and shuffle their order and train a new model. With a shuffled dataset, we obtained far lower BLEU scores of 12.84 and 16.73 for DE→HSB and HSB→DE respectively. As these BLEU scores are lower than even the scores obtained via sentence-level selection, we can conclude that the XLM model greatly benefits from sentences being in order for pretraining. However, it does appear that sentence-level selection provides higher quality sentences individually.

3.3 Quality versus quantity

With both selection methods, we can choose a threshold to determine how many sentences we should use for training our model. We start by selecting roughly 93 million sentences from News Crawl 2007-2019.[8] We chose the first 10 million sentences from each year, apart from 2008 and 2009, which only contain roughly 6.5 million sentences each. The sentences are chosen at the document-level. From the 93 million sentences combined, we use document-level selection to choose various amounts of data, varying from 1 million to 20 million sentences, and train models on each. The results are shown in Table 2.

As we can see, selecting 5 million sentences results in the highest BLEU scores. As data is either added or removed, the performance drops by around 1-2 BLEU. Given the nature of attention-based neural models, it is somewhat surprising to see that using more data is not helpful and in fact potentially harmful. Whether this is a peculiarity of the German–Upper Sorbian data or not requires further investigation.

Sentences (M)	DE→HSB	HSB→DE
1	16.01	17.14
2	15.20	16.61
5	**17.18**	**19.32**
10	16.78	18.65
20	16.09	17.75

Table 2: BLEU scores of models trained on varying amounts of document-level selected data.

Sentences (M)	DE→HSB	HSB→DE
2	17.76	19.19
5	**18.04**	**19.57**

Table 3: BLEU scores of models trained using 5 million sentences from News Crawl and various amounts of sentences from Common Crawl.

3.4 Using Common Crawl data

As a portion of the Upper Sorbian data is crawled from the web, we also perform data selection on Common Crawl. Since document boundaries are not available for Common Crawl, we can only use sentence-level selection. [9] We tested using various amounts of data in addition to the 5 million News Crawl sentences and report results in Table 3.

As we can see the system with 5 million News Crawl sentences and 5 million Common Crawl sentences performed the best. While the improvements are marginal, this may be due to a similar phenomenon as in Table 2, where too much monolingual data is not beneficial.

3.5 Iterative data selection

Since we saw improvements from one round of data selection, it would stand to reason that using a more accurate model to translate the Upper Sorbian data to German would result in potentially better data selection. As such, we use our model trained on 5 million sentences selected from News Crawl to translate the Upper Sorbian data into German, and apply the same data selection process on the roughly 93 million sentences as before.

The results on the second iteration are markedly worse, with BLEU scores of 15.9 and 17.45, on DE→HSB and HSB→DE, respectively, compared

[8] We exclude years 2007, 10, and 14 as they are used for training our initial model and thus may affect the selection.

[9] Our finding that randomly selected sentences indeed perform better was done post-hoc, which is why we use sentences selected with the highest scores.

to the original scores of 17.18 and 19.32. We suspect that this is due to the same data being used for training the NMT system and for selection, despite the data being used to train the KenLM models being different.[10]

This highlights a major downside of data selection using our methods: data cannot be used both for training a selection model and for the selection itself. The most likely reason for this is that the model will give all sentences that appear in the original training set higher scores, and documents which include the same or similar sentences will be chosen over documents that are more unique, effectively leading to an overfitting problem. This then raises a question of trade-off: is it better to use worse quality data to train the initial model and to then select from better quality data, or vice versa? Our results seem to indicate the former, but further research is required to get a definitive answer.

3.6 Further Analysis

To further analyze the data selected by the model, we look at the frequencies of words that appear in the selected data. We compare our document-filtered data from Section 3.1 with the data from the Upper Sorbian side for 10 word roots in Table 4. These word roots are selected manually as the correctly translated root is easy to verify (with Wikipedia and Wiktionary), and the translations are also one-to-one (ignoring the suffixes). We also select roots with varying frequency within the Upper Sorbian dataset.

As we can see, the high-quality document-filtered data has higher relative frequencies for the first 7 out of 10 word roots, and the lower-quality data has higher frequencies for the last 3. As the words are in order of frequency within the Upper Sorbian dataset, this indicates that the higher quality filtered data better represents the topics found in the Upper Sorbian dataset. Roots such as Sorbia- and Bautzen (a city where Sorbian is spoken) appear far more often in the higher quality data, despite being relatively uncommon in the German dataset. The last 3 words are relatively rare in the Upper Sorbian data, so it makes sense that the higher quality filtered data would have fewer occurrences of these words. Although most of the examples are related to locations, we do see that Domowin- (the root for Domowina, a non-profit organization) and Catholic- appear to show the same trends.

We also looked at the relative frequencies of the years 2000-2025 across our various models to see the effect of our filtering methods in matching the Upper Sorbian data according to year. We expect that the filtered German data with the frequency distribution most closely matching the frequency distribution of the Upper Sorbian data will have the strongest NMT performance. We show the results in Figure 1.

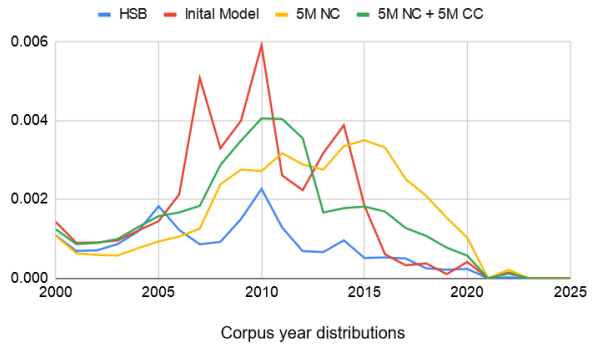


Figure 1: Relative frequencies of the years 2000-2025 within the various datasets. The frequencies are relative to the total number of sentences in that dataset.

Our initial model predictably has spikes in frequency at 2007, 2010, and 2014 as we manually chose data from these years to somewhat match the frequency of the Upper Sorbian data. Meanwhile, the 5 million document-level selected sentences from News Crawl seems to more closely match the frequencies in the Upper Sorbian data from 2000 to 2010, but has larger relative frequencies for years 2010 to 2020. We suspect that this is due to the limitation of the data available for selection, as earlier years have fewer sentences for the selection model to choose. Finally, the model using 5 million News Crawl and 5 million Common Crawl sentences has a frequency graph that most closely matches the graph of the Upper Sorbian data. The similarity of the Upper Sorbian graph to the other graphs seems to correlate with the resulting BLEU scores of the NMT model.

4 Conclusion

In the UNMT setting where one has access to a wealth of resources for one language, we investigated the feasibility of data selection. We attempt both document-level and sentence-level selection,

[10] We also saw similar performance drops when trying to include the data from years 2007, 10, and 14 in our original model trained on selected data, as these years were used to train the initial system used for selection.

Root			Count		Frequency %	
EN	DE	HSB	HSB	DE	Doc Low	Doc High
Sorbia-	Sorb-	Serbsk-	66105	187	0	97.3
German-	Deutsch-	Němsk-	17070	445203	18	21.8
Bautzen	Bautzen	Budyšin	11015	212	8.5	50.5
Lusatia-	Lausitz	Łužic-	10170	633	2.8	51.8
Domowin-	Domowin-	Domowin-	7835	32	0	100
Saxon-	Sachsen-	Saksk-	5163	10861	14.4	24.8
Catholic-	Kathol-	Katolsk-	4530	8515	12.7	28.9
Asia-	Asi-	Azij-	735	12175	31.4	11.3
Africa-	Afrik-	Afrik-	512	9967	23.2	15.7
Iran-	Iran-	Iran-	199	26714	53.9	4.6

Table 4: Frequencies of word roots within the Upper Sorbian (HSB), and relative frequencies of low-quality document-filtered (Doc Low) and high-quality document-filtered (Doc High) datasets. Relative frequency is based on the total frequency of each root within the 10 million sentences that the sets are selected from (i.e. the DE count column). Case is ignored when determining frequency.

finding that both methods are capable of distinguishing low quality data from high quality data, with quality in this case defined as the efficacy for training an XLM model. We found that while document-level selection chooses poorer sentences on average, the XLM model can leverage the inter-sentence information to achieve better results than when simply using the highest quality sentences. We also found that there appears to be a point where adding more monolingual data is not beneficial, but rather potentially harmful, indicating a need for data selection. Finally, we noted some potential drawbacks to using this form of data selection, particularly that data cannot be used for both initial training of the NMT model and subsequent selection. Future work could continue along many avenues, such as the effectiveness of data selection on other language pairs, or even on the Upper Sorbian side.

References

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems*, pages 7059–7069.

Kenneth Heafield, Ivan Pouzyrevsky, Jonathan H Clark, and Philipp Koehn. 2013. Scalable modified kneser-ney language model estimation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 690–696.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 177–180, Stroudsburg, PA, USA. Association for Computational Linguistics.

Robert C. Moore and William Lewis. 2010. Intelligent selection of language model training data. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 220–224, Uppsala, Sweden. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

The LMU Munich System for the WMT20 Very Low Resource Supervised MT Task

Jindřich Libovický and **Viktor Hangya** and **Helmut Schmid** and **Alexander Fraser**
Center for Information and Language Processing
LMU Munich
{libovicky,hangya,schmid,fraser}@cis.lmu.de

Abstract

We present our systems for the WMT20 Very Low Resource MT Task for translation between German and Upper Sorbian. For training our systems, we generate synthetic data by both back- and forward-translation. Additionally, we enrich the training data with German-Czech translated from Czech to Upper Sorbian by an unsupervised statistical MT system incorporating orthographically similar word pairs and transliterations of OOV words. Our best translation system between German and Sorbian is based on transfer learning from a Czech-German system and scores 12 to 13 BLEU higher than a baseline system built using the available parallel data only.

1 Introduction

In this paper, we describe systems for translation between German and Upper Sorbian developed at LMU Munich for the WMT20 shared task on very low-resource supervised MT.

Upper Sorbian is a minority language spoken by around 30,000 people in today's German state of Saxony. With such a small number of speakers, machine translation and automatic processing of Sorbian is an inherently low-resource problem without any chance that the resources available for Sorbian would ever approach the size of resources for languages spoken by millions of people. On the other hand, being a Western Slavic language related to Czech and Polish, it is possible to take advantage of relatively rich resources collected for these two languages.

The German-Sorbian systems presented in this paper are neural machine translation (NMT) systems based on the Transformer architecture (Vaswani et al., 2017). We experiment with various data preparation and augmentation techniques: back-translation (Sennrich et al., 2016b), finetuning systems trained for translation between Czech and German (Kocmi and Bojar, 2018), and data augmentation by including German-Czech parallel data with the Czech side translated to Upper Sorbian by an unsupervised system that includes an unsupervised transliteration model for guessing how to translate out-of-vocabulary Czech words to Upper Sorbian.

Our experiments show the importance of data augmentation via stochastic pre-processing and synthetic data generation. The best systems were trained by transfer-learning from a Czech-German system. However, compared to data augmentation, transfer learning from Czech-German translation only produces a minor improvement. Based on the preliminary shared task results, the presented systems scored on the 4th place among 10 competing teams in the shared task.

2 Related Work

Until recently, phrase-based approaches were believed to be more suitable for low-resource translation. Koehn and Knowles (2017) claimed that a parallel dataset of at least 10^7 tokens is required for NMT to outperform phrase-based MT. This view was also supported by the results of Artetxe et al. (2018b) and Lample et al. (2018), who showed that phrase-based approaches work well for unsupervised MT, at least in the early stages of the iterative back-translation procedure.

Recently, Sennrich and Zhang (2019) revisited the claims about data needs of supervised NMT and showed that with recent innovations in neural network and careful hyper-parameter tuning, NMT models outperform their phrase-based counterparts with training data as small as 100k tokens (15 times smaller than the data provided for this shared task).

Standard techniques for low-resource machine translation include data augmentation with rule-based substitutions (Fadaee et al., 2017), by sam-

Proceedings of the 5th Conference on Machine Translation (WMT), pages 1104–1111
Online, November 19–20, 2020.

Data		# sent.	# tok.	$\frac{\text{\# tok.}}{\text{\# sent.}}$
Train	de	60k	822k	13.7
	hsb		738k	12.3
Devel	de	2k	28k	13.8
	hsb		25k	12.5
Devel test	de	2k	28k	13.9
	hsb		25k	12.7
German-Czech newstest2019	de	2k	49k	24.5
	cs		43k	22.0
German-Czech parallel	de	14.7M	234M	15.9
	cs		219M	14.8

Table 1: Statistics on the parallel data compared to German-Czech News Test 2019 and parallel German-Czech data (see Section 3.3).

pling synthetic noise (Wang et al., 2018; Provilkov et al., 2020), or by iterative back-translation (Hoang et al., 2018). Another class of approaches relies on transfer learning from models trained for high-resource language pairs of more or less similar languages (Zoph et al., 2016; Nguyen and Chiang, 2017; Kocmi and Bojar, 2018).

3 Data

We used several types of data to train our systems. The organizers provided authentic parallel data and Sorbian monolingual data. We also use German and Czech News Crawl data and Czech-German parallel data available in Opus (Tiedemann, 2012).

3.1 Authentic Parallel Data

The organizers of the shared task provided a parallel corpus of 60k sentences, and validation and development test data of 2k sentences each.

The basic statistics about the data are presented in Table 1. Note that the sentences are on average much shorter and therefore also likely to be structurally simpler than in the type of sentences usually used in the WMT test sets.

3.2 Monolingual Data

In total 696k monolingual Sorbian sentences were provided by the organizers. We noticed that the monolingual Sorbian data contain many OCR-related errors originating from hyphenation. We thus removed all sentences ending with a hyphen. Additionally, we merged tokens ending with a hyphen with the adjacent one if such merging results in a known Sorbian word. This filtered out 1.6k sentences and did 12k token merges.

The monolingual Sorbian data were used for training the unsupervised Czech-Sorbian translation system (see Section 4.1) and for back-translation in Sorbian-German systems.

Besides, we use 60M German and 60M Czech sentences from the NewsCrawl data provided as monolingual data for WMT shared tasks (Barrault et al., 2019). The monolingual data were used for generating synthetic training data via back- and forward-translation both for the German-Sorbian and German-Czech systems. In addition, the Czech monolingual data was used in the unsupervised Czech-Sorbian translation system as well.

3.3 German-Czech Data

For transfer learning and the creation of synthetic data, we also used German-Czech parallel data. We downloaded all available parallel datasets from the Opus project (Tiedemann, 2012), which gave us 20.8M parallel sentences, which we further filtered.

First, we filtered the parallel sentences by length. We estimated the mean and the standard deviation of the length ratio of German and Czech sentences and kept only those sentence pairs whose length ratio fitted into the interval of two times standard deviation around the mean. Then, we applied a language identifier from FastText (Grave et al., 2018) and only kept sentence pairs identified as German-Czech. The filtering lefts us with 14.7M parallel sentences.

4 Synthetic data from Czech-German

Since Upper Sorbian is related to Czech, we generate additional synthetic parallel German-Sorbian data by translating the Czech side of the German-Czech parallel data. For this, we use an unsupervised statistical MT system which includes mined Czech-Sorbian transliteration word pairs for better performance.

4.1 Unsupervised SMT

We follow the approach of Artetxe et al. (2018b) to build an Unsupervised Statistical Machine Translation (SMT) system. In the following description, we mainly focus on the steps that we changed compared to the original system and keep the description of the other steps brief.

In the first step, we build 300-dimensional monolingual n-gram embeddings for both Czech and Sor-

bian using *FastText skip-gram* (Bojanowski et al., 2017) on the above mentioned monolingual data. We restrict the vocabulary to the most frequent 200k, 400k, and 400k 1-, 2- and 3-grams, respectively. We map these embeddings to a shared bilingual space using *VecMap* (Artetxe et al., 2018a). In contrast to the original unsupervised SMT pipeline, which builds bilingual word embeddings (BWEs) without any cross-lingual signal, we use identical words occurring in both languages as the seed lexicon for the mapping. We found that the available small monolingual Sorbian corpus is not adequate to build BWEs in a fully unsupervised way. The corpora are tokenized and true-cased using *Moses* tools (Koehn et al., 2007). We note that because there are no available language rules for Sorbian, we used Czech rules for tokenization, which is reasonable because of the similarity of the two languages.

We build phrase tables for both translation directions. For each source n-gram, we take 100 candidates with the closest embeddings based on cosine similarity and additional 100 candidates with the smallest edit distance. We calculate 5 scores for each pair: phrase and lexical translation probabilities and their inverse as in (Artetxe et al., 2018b), and their normalized edit distance. For phrases, the latter is calculated by pairing each source word with the most similar target side word and taking the average edit distance of each of these pairs as the normalization constant. In addition to the phrase tables, we train language models using the monolingual corpora.

We use the validation set from the shared task (with the German side machine-translated to Czech) to tune the parameters with MERT instead of tuning on synthetic data. Finally, we run 3 iterative refinement steps.

4.2 Translating OOVs by Transliteration

Because of the small monolingual data, the Sorbian vocabulary is relatively small. To improve on this problem, we exploit the similarity of Upper Sorbian and Czech by translating Czech out-of-vocabulary (OOV) words to Upper Sorbian, using transliteration. More precisely, we transliterate Czech words from the German-Czech parallel data which were not seen by the SMT system during training, assuming that the translations of these words are missing in the Sorbian vocabulary on the target side as well. We extracted the training data for the transliteration system using a preliminary *transliteration mining* model, *filtered* the data using a preliminary transliteration model, and trained the final *transliteration model* on the filtered data.

Transliteration mining. Our transliteration mining is similar to the model by Sajjad et al. (2012). It consists of a transliteration submodel and a noise submodel.

The *transliteration submodel* is a unigram model over transliteration units (TUs) which jointly generates a source and a target language string. The English-German transliteration pair *(Gorbatchev, Gorbatschow)* could be generated as the following sequence of TUs: G:G o:o r:r b:b a:a t:t s: c:c h:h e:o v:w. We use only 1-1, 0-1, and 1-0 TUs. The probability $\mathbf{p}(\mathbf{a})$ of a sequence of TUs is the product of the unigram probabilities $p(a_i)$:

$$p(\mathbf{a}) = p(a_1, ..., a_n) = \prod_{i=1}^{n} p(a_i)$$

Probability $p_{\text{trans}}(\mathbf{s}, \mathbf{t})$ of a string pair is obtained by summing over all possible alignments $\mathbf{a}$:

$$p_{\text{trans}}(\mathbf{s}, \mathbf{t}) = \sum_{\mathbf{a} \in \text{align}(\mathbf{s}, \mathbf{t})} p(\mathbf{a})$$

The *noise submodel* independently generates a source string $\mathbf{s}$ and a target string $\mathbf{t}$ using two unigram models over the source and the target language characters, respectively. The probability of a string pair is the product of the two monolingual string probabilities:

$$p_{\text{noise}}(\mathbf{s}, \mathbf{t}) = p_{\text{src}}(\mathbf{s})\ p_{\text{tgt}}(\mathbf{t})$$

The monolingual probability of the source string (and analogously the target string) is defined as a product of letter unigram probabilities.

Sajjad et al. (2012) *interpolate* the noise model and the target model as a linear combination. Unfortunately, such a model also extracts near-transliterations which differ from a true transliteration by e.g., an inflexional affix, such as *(Gorbatchev, Gorbatschows)*.

Instead, we combine the two submodels by *concatenation*. Our model produces a word pair by (i) generating two word prefixes[1] $\mathbf{s}^p$ and $\mathbf{t}^p$ using the noise model, (ii) generating two middle parts $\mathbf{s}^m$ and $\mathbf{t}^m$ using the transliteration model, and

[1] Here, the terms *prefix* and *suffix* are not used in a linguistic sense.

(iii) generating two suffixes $\mathbf{s}^s$ and $\mathbf{t}^s$ using the noise model. The intuition is that if the most probable way to generate a pair does not use prefixes or suffixes, it is a transliteration. Here the non-transliteration pair *(Gorbatchev, Gorbatschows* might be most probably obtained by generating two empty strings as prefixes with the noise submodel, the TU sequence G:G o:o r:r b:b a:a t:t s: c:c h:h e:o v:w with the transliteration submodel, and an empty suffix and the suffix *s* with the noise model.

The probability is defined as follows:

$$p(\mathbf{s^p}, \mathbf{s^m}, \mathbf{s^s}, \mathbf{t^p}, \mathbf{t^m}, \mathbf{t^s}) = p_{\text{noise}}(\mathbf{s^p}, \mathbf{t^p}) p_{\text{trans}}(\mathbf{s^m}, \mathbf{t^m}) p_{\text{noise}}(\mathbf{s^s}, \mathbf{t^s})$$

The total probability of a word pair is obtained by summing over all possible splits:

$$p(\mathbf{s}, \mathbf{t}) = \sum_{\substack{\mathbf{s^p}, \mathbf{s^m}, \mathbf{s^s}, \mathbf{t^p}, \mathbf{t^m}, \mathbf{t^s} \\ \in \text{split}(\mathbf{s}, \mathbf{t})}} p(\mathbf{s^p}, \mathbf{s^m}, \mathbf{s^s}, \mathbf{t^p}, \mathbf{t^m}, \mathbf{t^s})$$

The parameters of the transliteration submodel are trained using the EM algorithm on the list of transliteration candidates. The parameters of the monolingual models are estimated directly from the data and kept fixed during training. After the EM training, we compute for each candidate pair, the most probable split of the two words into prefix/middle/suffix, and the most probable alignment of the two middle parts using the Viterbi algorithm. If all prefixes and suffixes are empty, the candidate pair is extracted as a probable transliteration.

We run the transliteration mining on lower-cased data and consider all possible word pairs with a reasonable edit distance. The mining process returns the extracted transliteration candidates and their most probable TU sequence, respectively.

Transliteration filtering. The mining process only relies on the unigram probabilities, which is often suboptimal. Therefore, we add a filtering step that scores each transliteration pair using an n-gram transliteration model and eliminates pairs with a low score.

We train a Kneser-Ney-smoothed trigram transliteration model on the TU sequences of the transliterations extracted using the transliteration mining model.

For each extracted transliteration pair, we compute negative log probabilities:

- L_1 of the corresponding TU sequence;
- L_2 of the best source-to-target transliteration; and
- L_3 of the best target-to-source transliteration.

We filter out a word pair if $L_2 - L_1 + L_3 - L_1 > 10$. Note that all three probabilities are joint probabilities and that the same transliteration model can be used in both directions.

Unsupervised SMT	12.0
+ edit distance	13.3
+ transliteration	13.8

Table 2: BLEU scores of the Czech-Sorbian system with gradually added techniques measured on the Upper Sorbian-German test set where the German side has been machine-translated to Czech.

Transliteration Generation. We train the final transliteration model on the TU sequences of the filtered transliteration pairs and use the model to generate Sorbian transliterations for Czech OOV words. We lowercase the Czech words before transliteration and transfer the casing from the original Czech words to their Sorbian transliterations.

Using the model, we generate transliterations for Czech words not seen by the unsupervised SMT system during training, i.e., we take all the words from the Czech side of the parallel data which are not present in the used Czech monolingual corpus. To add these word pairs to the SMT system, we consider them as a parallel corpus and concatenate it to the synthetic parallel data created in the iterative refinement steps and also update the language models. We run two additional refinement steps on top of the three mentioned in 4.1. Finally, we create the synthetic German-Sorbian data by translating the Czech side of the German-Czech data and feed it to our final NMT system, as described below.

Table 2 shows the translation quality of the unsupervised SMT system. The basic setup relies only on BWEs to build the initial phrase tables. Next, we add edit distance information, and finally, we use the mined transliteration pairs as well. However, note that the BLEU scores are very approximate because the source side of the test is machine-translated.

5 Experimental Setup

For the translation between German and Sorbian, we experimented with NMT models based on

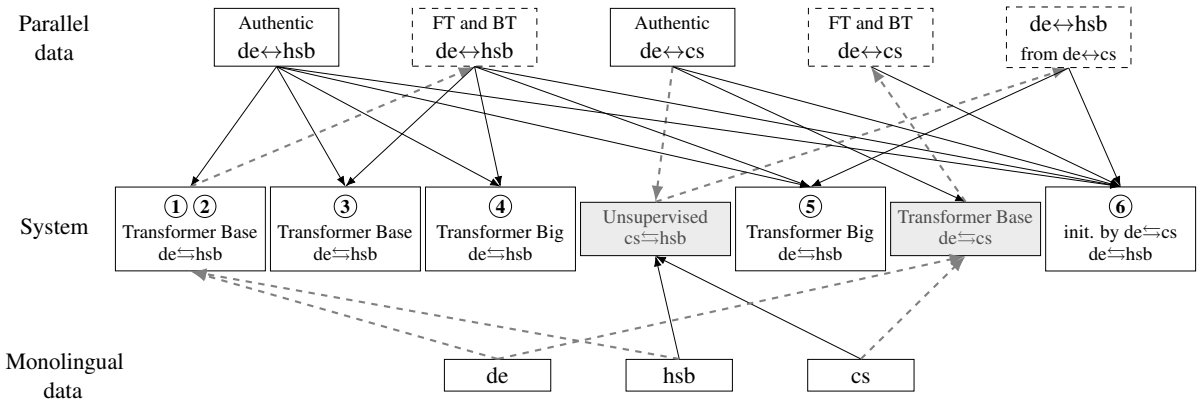


Figure 1: Overview of datasets and systems that were used to generate synthetic data. Solid arrows denote training a system, dashed gray arrows denote using the model for data generation. Synthetic datasets have dashed boxes.

Transformers (Vaswani et al., 2017). We followed known best practices for architecture and optimization choices. In our experiments, we mostly focus on data engineering.

5.1 Model Architecture and Optimization

We use the Transformer architecture (Vaswani et al., 2017) as implemented in Marian (Junczys-Dowmunt et al., 2018). For the initial experiments, we used the Base architecture (6 layers, hidden state of size 512, 8 attention heads, feed-forward layer 1024), and Big for the later experiments (12 layers, hidden state 1024, 16 attention heads, feed-forward layer 4096). We follow the default standard learning rate schedule proposed by Vaswani et al. (2017) with learning rate $3 \cdot 10^{-4}$. We use 16k warm-up steps for the Base architecture and 32k warm-up steps for the Big architecture.

The Base architecture is used for the initial systems which generate synthetic data via backward- and forward-translation. We use the Big architecture for the rest of the systems.

5.2 Training Data Preparation

An overview of the data generation and system training steps is provided in Figure 1.

We use a common BPE-based vocabulary (Sennrich et al., 2016c) for all systems which allows us to better ensemble our systems. Instead of proper tokenization, we use the pre-tokenization heuristic from SentencePiece (Kudo and Richardson, 2018) as implemented in YouTokenToMe.[2] The BPE vocabulary consists of 16k merges and was fit using the authentic parallel training data only.

[2]https://github.com/VKCOM/YouTokenToMe

We apply BPE-dropout (Provilkov et al., 2020) of 0.1 on both the source and the target side of the data. We oversample the monolingual data 1000 times and with different segmentations (Model 2). We hypothesize that in the very low-resource setup, the BPE dropout serves more as a data-augmentation technique than as regularization.

Due to hardware limitations, we limit the data mixes for training the Big architectures to 180M parallel sentences. One third of the data mix consists of oversampled authentic parallel data. In one set of experiments (Models 3, 4), the rest of the data consists of synthetic data: an equal number of samples of forward- and back-translation (which means that the monolingual Sorbian data is oversampled approximately 80×). In another set of experiments (Model 5), we additionally sample data from the machine-translated Czech-German data set where the Czech part has been automatically translated to Upper Sorbian. Following Caswell et al. (2019), we tag the synthetic data, having a separate tag for each of the synthetic data types.

Further, we experiment with finetuning models originally trained for translation between Czech and German. The data for the parent models is prepared using the same protocol as for Model 4. Following Kocmi and Bojar (2018), we train the parent model until convergence and continue training with the German-Sorbian data. Based on preliminary results, we use the data mix for Model 4 for the German-to-Sorbian translation direction and the data mix for Model 5 for translating from Sorbian into German.

	Model	hsb→de		de→hsb	
1	Transformer Base, parallel only	43.4	.695	45.6	.702
2	(1) + BPE dropout	50.9	.745	51.7	.747
3	(2) + back- and forward-translation	51.6	.766	52.4	.765
4	Transformer Big, same data as (3)	53.0	.766	55.3	.765
5	(4) + synthetic data from cs-de	54.2	.766	54.9	.766
6	(4/5) initialized by cs⇄de	55.4	.772	55.9	.775
7	Ensemble 4× (4/5)	55.0	.772	55.9	.773
8	Ensemble 3× (6)	55.6	.773	56.2	.776
9	Ensemble 4× (4/5) and 3× (6)	56.0	.777	56.9	.769
10	(4/5) trained right-to-left	53.7	.765	55.1	.769
11	(9) + right-to-left rescoring	56.0	.778	57.0	.779

Table 3: BLEU scores and chrF scores (in small font) on development test data for Sorbian-to-German (hsb→de) and German-to-Sorbian (de→hsb) translations.

5.3 Model Ensembling

Following Sennrich et al. (2016a), we also experiment with ensembling several systems and combining systems trained in the left-to-right and right-to-left direction.

We trained four models from random initialization and three models by transferring from Czech-German translation. Note that the transferred models were initialized by the same model and only differed in the order of the training data.

Further, we trained two models in the right-to-left direction, starting from random initialization.

6 Results

The quantitative results in terms of BLEU score (Papineni et al., 2002) and ChrF (Popović, 2017) score are presented in Table 3. The results were measured using SacreBLEU.[3]

The Base architecture trained using the parallel data only (Model 1) reaches a surprisingly high BLEU score, which is probably due to the quality of the manually curated training data, domain closeness of the train and test data, and relatively simple sentences both in the train and test sets.

The data augmentation using BPE-dropout (Model 2) seems to have a substantial effect on the translation quality, improving the translation by 6–7 BLEU points. This is a much larger effect than Provilkov et al. (2020) reported. However, they also observed a larger positive effect on smaller datasets. Unlike Sennrich and Zhang (2019), we did not find any benefits of using a small BPE-based vocabulary or tuning learning rate. However, the positive effect of the small vocabulary might be partially emulated by the BPE dropout.

Adding the back- and forward-translated data in the training data improved the translation quality only slightly (Model 3). A large positive effect can be achieved by switching to the Big architecture (Model 4). Adding the synthetic data generated from Czech-German parallel data improve only the Sorbian-to-German translation direction (Model 5), presumably because the quality of the synthetic Sorbian side of the corpus is too low to be used as a target side.

Transfer learning from German-Czech models further improves the translation quality by approximately 1 BLEU point. These are thus the best single models we have developed and our contrastive submission to the shared task.

Additional improvements were reached by model ensembling. Ensembling both the model trained from random initialization and transfer learning models improves the translation by approx. 1 BLEU point. Ensembling these two model types together further improves the translation quality by around half BLEU point.

The model generating the translation right-to-left reach translation quality that is comparable to the left-to-right models. However, rescoring of the n-best lists generated by left-to-right ensembles by the right-to-left models improves the translation quality only negligibly. The rescored ensemble was our primary submission to the shared task.

[3] https://github.com/mjpost/sacrebleu

7 Conclusions

We presented NMT systems for translation between German and Upper Sorbian. Due to the domain closeness and relative simplicity of the test data, we were able to achieve BLEU scores over 50 using the parallel data only. The crucial component was the use of BPE-dropout for both the source and target side.

Further translation quality improvements were achieved by generating synthetic training data by back- and forward-translation. Additionally, we generated synthetic data by machine-translating the Czech side of a parallel German-Czech corpus. For that, we built an unsupervised SMT system that additionally utilizes an unsupervised transliteration system for the translation of OOV tokens.

Our best single system is based on transfer learning, i.e., initializing the model by a Czech-German system, reaching 1–2 higher BLEU scores compared to systems based on Sorbian and German data only. Further minor improvements were achieved by model ensembling and right-to-left rescoring.

Acknowledgments

The work was supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 640550) and by German Research Foundation (DFG; grant FR 2829/4-1).

References

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018a. A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 789–798, Melbourne, Australia. Association for Computational Linguistics.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018b. Unsupervised statistical machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3632–3642, Brussels, Belgium. Association for Computational Linguistics.

Loïc Barrault, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. 2019. Findings of the 2019 conference on machine translation (WMT19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 1–61, Florence, Italy. Association for Computational Linguistics.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Isaac Caswell, Ciprian Chelba, and David Grangier. 2019. Tagged back-translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 53–63, Florence, Italy. Association for Computational Linguistics.

Marzieh Fadaee, Arianna Bisazza, and Christof Monz. 2017. Data augmentation for low-resource neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 567–573, Vancouver, Canada. Association for Computational Linguistics.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*, Miyazaki, Japan. European Languages Resources Association (ELRA).

Vu Cong Duy Hoang, Philipp Koehn, Gholamreza Haffari, and Trevor Cohn. 2018. Iterative back-translation for neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 18–24, Melbourne, Australia. Association for Computational Linguistics.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia. Association for Computational Linguistics.

Tom Kocmi and Ondřej Bojar. 2018. Trivial transfer learning for low-resource neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 244–252, Belgium, Brussels. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion*

Volume Proceedings of the Demo and Poster Sessions, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39, Vancouver. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Phrase-based & neural unsupervised machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5039–5049, Brussels, Belgium. Association for Computational Linguistics.

Toan Q. Nguyen and David Chiang. 2017. Transfer learning across low-resource, related languages for neural machine translation. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 296–301, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Maja Popović. 2017. chrF++: words helping character n-grams. In *Proceedings of the Second Conference on Machine Translation*, pages 612–618, Copenhagen, Denmark. Association for Computational Linguistics.

Ivan Provilkov, Dmitrii Emelianenko, and Elena Voita. 2020. BPE-dropout: Simple and effective subword regularization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1882–1892, Online. Association for Computational Linguistics.

Hassan Sajjad, Alexander Fraser, and Helmut Schmid. 2012. A statistical model for unsupervised and semi-supervised transliteration mining. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 469–477, Jeju Island, Korea. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Edinburgh neural machine translation systems for WMT 16. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 371–376, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016c. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich and Biao Zhang. 2019. Revisiting low-resource neural machine translation: A case study. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 211–221, Florence, Italy. Association for Computational Linguistics.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, pages 2214–2218, Istanbul, Turkey. European Language Resources Association (ELRA).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 5998–6008.

Xinyi Wang, Hieu Pham, Zihang Dai, and Graham Neubig. 2018. SwitchOut: an efficient data augmentation algorithm for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 856–861, Brussels, Belgium. Association for Computational Linguistics.

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer learning for low-resource neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1568–1575, Austin, Texas. Association for Computational Linguistics.

NRC Systems for Low Resource German-Upper Sorbian Machine Translation 2020: Transfer Learning with Lexical Modifications

Rebecca Knowles and **Samuel Larkin** and **Darlene Stewart** and **Patrick Littell**
National Research Council Canada
{Rebecca.Knowles, Samuel.Larkin, Darlene.Stewart, Patrick.Littell}@nrc-cnrc.gc.ca

Abstract

We describe the National Research Council of Canada (NRC) neural machine translation systems for the German–Upper Sorbian supervised track of the 2020 shared task on Unsupervised MT and Very Low Resource Supervised MT. Our models are ensembles of Transformer models, built using combinations of BPE-dropout, lexical modifications, and backtranslation.

1 Introduction

We describe the National Research Council of Canada (NRC) neural machine translation systems for the shared task on Unsupervised MT and Very Low Resource Supervised MT. We participated in the supervised track of the low resource task, building Upper Sorbian–German neural machine translation (NMT) systems in both translation directions. Upper Sorbian is a minority language spoken in Germany. We built baseline systems (standard Transformer (Vaswani et al., 2017) with a byte-pair encoding vocabulary (BPE; Sennrich et al., 2016b)) trained on all available parallel data (60,000 lines), which resulted in unusually high BLEU scores for a language pair with such limited data.

In order to improve upon this baseline, we used transfer learning with modifications to the training lexicon. We did this in two ways: by experimenting with the application of BPE-dropout (Provilkov et al., 2020) to the transfer learning setting (Section 2.3), and by modifying Czech data used for training parent systems with word and character replacements in order to make it more "Upper Sorbian-like" (Section 2.4).

Our final systems were ensembles of systems built using transfer learning and these two approaches to lexicon modification, along with iterative backtranslation.

2 Approaches

2.1 General System Notes

In both translation directions, our final systems consist of ensembles of multiple systems, built using transfer learning (Section 2.2), BPE-Dropout (Section 2.3), alternative preprocessing of Czech data (Section 2.4), and backtranslation (Section 2.5). We describe these approaches and related work in the following sections, providing implementation details for reproducibility in Sections 3, 4 and 5.

2.2 Transfer Learning

Zoph et al. (2016) proposed a transfer learning approach for neural machine translation, using language pairs with larger amounts of data to pre-train a parent system, followed by finetuning a child system on the language pair of interest. Nguyen and Chiang (2017) expand on that, showing improved performance using BPE and shared vocabularies between the parent and child. We follow this approach: we build disjoint source and target BPE models and vocabularies, with one vocabulary for German (DE) and one for the combination of Czech (CS) and Upper Sorbian (HSB); see Section 4.

We chose to use Czech–German data as the parent language pair due to the task suggestions, relative abundance of data, and the close relationship between Czech and Upper Sorbian (cf. Lin et al., 2019; Kocmi and Bojar, 2018). While Czech and Upper Sorbian cognates are often not identical at the character level (Table 1), there is a high level of character-level overlap; trying to take advantage of that overlap without assuming complete character-level identity is a motivation for the explorations in subsequent sections (Section 2.3, Section 2.4). Another relatively high-resource language related to Upper Sorbian is Polish, but while the Czech and Upper Sorbian orthographies are fairly similar, mostly using the same characters for the same

Proceedings of the 5th Conference on Machine Translation (WMT), pages 1112–1122

sounds (with a few notable exceptions), Polish orthography is more distinct. This, combined with the lack of a direct Polish–German parallel dataset in the constrained condition, led us to choose Czech as our transfer language for these experiments.

Czech	Upper Sorbian
analyzovat	analyzować
donesl	donjesł
externích	eksternych
hospodářská	hospodarsce
kreativní	kreatiwne
okres	wokrjes
potom	potym
projekt	projekt
sémantická	semantisku
velkým	wulkim

Table 1: A sample of probable Czech–Upper Sorbian cognates and shared loanwords, mined from the Czech–German and German–Upper Sorbian parallel corpora and filtered by Levenshtein distance.

Other work on transfer learning for low-resource machine translation includes multilingual seed models (Neubig and Hu, 2018), dynamically adding to the vocabulary when adding languages (Lakew et al., 2018), and using a hierarchical architecture to use multiple language pairs (Luo et al., 2019).

2.3 BPE-Dropout

We apply the recently-proposed approach of performing BPE-dropout (Provilkov et al., 2020), which takes an existing BPE model and randomly drops some merges at each merge step when applying the model to text. The goal of this, beside leading to more robust subword representations in general, is to produce subword representations that are more likely to overlap between the pretraining (Czech–German) and finetuning (Upper Sorbian–German) stages. We hypothesized that, in the same way that BPE-Dropout leads to robustness against accidental spelling errors and variant spellings (Provilkov et al., 2020), it could likewise lead to robustness to the kind of spelling variations we see between two related languages.

For example, consider the putative Czech–Upper Sorbian cognates and shared loanwords presented in Table 1. Sometimes a fixed BPE segmentation happens to separate shared characters into shared subwords (e.g. CS `analy@@ z@@ ovat` vs. HSB `analy@@ z@@ ować`), such that the presence of the former during pre-training can initialize at least some of the subwords that the model will later see in Upper Sorbian. However, other times the character-level differences lead to segmentations where no subwords are shared (e.g. CS `hospodář@@ ská` vs. HSB `hospodar@@ sce` or `potom` vs. HSB `po@@ tym`). Considering a wider variety of segmentations would, we hypothesized, mean that Upper Sorbian subwords would have more chance of being initialized during Czech pre-training (see Appendix C).

Rather than modifying the NMT system itself to reapply BPE-dropout during training, we treated BPE-dropout as a preprocessing step. Additionally, we experimented with BPE-dropout in the context of transfer learning, examining the effects of using source-side, both-sides, or no dropout in both parent and child systems.

2.4 Pseudo-Sorbian

For the Upper Sorbian–German direction, we also experimented with two techniques for modifying the Czech–German parallel data so that the Czech side is more like Upper Sorbian. In particular, we concentrated on modification methods that require neither large amounts of data, nor in-depth knowledge of the historical relationships between the languages, since both of these are often lacking for the lower-resourced language.

We considered two variations of this idea:

- *word-level* modification, in which some frequent Czech words (e.g. prepositions) are replaced by likely Upper Sorbian equivalents, and
- *character-level* modification, where we attempt to convert Czech words at the character level to forms that may more closely resemble Upper Sorbian words.

Note that in neither case do we know what particular conversions are *correct*; we ourselves do not know enough about historical Western Slavic to predict the actual Upper Sorbian cognates of Czech words. Rather, we took inspiration from stochastic segmentation methods like BPE-Dropout (Provilkov et al., 2020) and SentencePiece (Kudo and Richardson, 2018): when we have an idea of the *possible* solutions to the segmentation problem but do not know which one is the *correct* one, we can sample randomly from the possible segmentations as a sort of regularization, with the

goal of discouraging the model from relying too heavily on a single segmentation scheme and giving it some exposure to a variety of possible segmentations. Whereas BPE-dropout and SentencePiece focus on possible segmentations of the word, our pseudo-Sorbian experiments focus on possible word- and character-level replacements. The goal was to discourage the parent Czech–German model from relying too heavily on regularities in Czech (e.g. the presence of particular frequent words, the presence of particular Czech character *n*-grams) and perhaps also gain some prior exposure to Upper Sorbian words and characters that will occur in the genuine Upper Sorbian data; we can also think of this as a form of low-resource data augmentation (Fadaee et al., 2017; Wang et al., 2018). See Appendix C for an analysis of increased subword overlap between pseudo-Sorbian and test data, as compared to BPE-dropout and the baseline approach.

2.4.1 Word-level pseudo-Sorbian

To generate the word-level pseudo-Sorbian, we ran `fast_align` (Dyer et al., 2013) on the Czech–German and German–Upper Sorbian parallel corpora, and took the product of the resulting word correspondences, to generate candidate Czech-Upper Sorbian word correspondences. As this process produces many unlikely correspondences, particularly for words that occur only a few times in the corpora, we filtered this list so that any Czech–German word correspondence that occurred fewer than 500 times in the aligned corpus was ineligible, and likewise any German–Upper Sorbian correspondence that occurred fewer than 50 times. We then used these correspondences to randomly replace 10% of eligible Czech words in the Czech-German corpus with one of their putative equivalents in Upper Sorbian. The result is a language that is *mostly* still Czech, but in which some high-frequency words (especially prepositions) are Upper Sorbian.

2.4.2 Character-level pseudo-Sorbian

To generate the character-level pseudo-Sorbian, we began with the same list of putative Czech-Upper Sorbian word correspondences, calculated the Levenshtein distances (normalized by length) between them, and filtered out pairs that exceeded 0.5 distance. This gave a list of words that were likely cognates, from which we hand-selected a development set of about 200; a sample of these is seen in Table 1. Using this set to identify character-level correspondences (e.g. CS `v` to HSB `w`, CS `d` to HSB `dź` before front vowels, etc.), we wrote a program to randomly replace the appropriate Czech character sequences with possible correspondences in Upper Sorbian. Again, as Czech-Upper Sorbian correspondences are not entirely predictable (CS `e` might happen to correspond, in a particular cognate, to HSB `e` or `ej` or `i` or `a` or `o`, etc.), we cannot expect that any given result is correct Upper Sorbian. Rather, we can think of this process as attempting to train a system that can respond to inputs from a variety of possible (but not necessarily actual) Western Slavic languages, rather than just a system that can respond to precisely-spelled Czech and only Czech.

2.4.3 Combined pseudo-Sorbian

In initial testing, we determined that a combination of word-level and character-level modification performed best; we ran each process on the Czech–German corpus separately, then concatenated the resulting corpora and trained a parent model on it. Due to time constraints we did not run the full set of ablation experiments. Subsequent finetuning on genuine Upper Sorbian–German data proceeded as normal, without any modification.

For all pseudo-Sorbian systems, we used the BPE vocabulary trained on the original Czech and Upper Sorbian data, rather than the modified data, so that systems trained on pseudo-Sorbian data could still be ensembled with systems trained only on the original data (Section 2.6).

2.5 Backtranslation

We used backtranslation (Sennrich et al., 2016a) to incorporate monolingual German and Upper Sorbian data into training. We backtranslated all Upper Sorbian monolingual data (after filtering as described in Section 3). We backtranslated the German monolingual news-commentary data and 1.2M randomly sampled lines of 2019 German news.

We experiment with iterative backtranslation: backtranslating data using systems without backtranslation, and then using the new systems built using the backtranslated text to perform a second iteration of backtranslation (Hoang et al., 2018; Niu et al., 2018; Zhang et al., 2018). Like Caswell et al. (2019), we use source-side tags at the start of backtranslated sentences to indicate to the models which sentences are the product of backtranslation.

2.6 Ensembling

Our final systems are ensembles of several systems. Because all systems used the same vocabulary sets and same model sizes, we could decode using Sockeye's (Hieber et al., 2018) default ensembling mechanism.

3 Data

We used all provided parallel German–Upper Sorbian data and all monolingual Upper Sorbian data (after filtering), along with German–Czech parallel data from Open Subtitles (Lison and Tiedemann, 2016),[1] DGT (Tiedemann, 2012; Steinberger et al., 2012), JW300 (Agić and Vulić, 2019), Europarl v10 (Koehn, 2005), News-Commentary v15, and WMT-News[2] for building the BPE vocabularies. The monolingual Upper Sorbian Web and Witaj datasets[3] were filtered to remove lines containing characters that had not been observed in the Upper Sorbian parallel data or in the Czech data; this removed sentences that contained text in other scripts and other languages. The Czech–German data was used for training parent models, while monolingual German and Upper Sorbian were used (along with parallel German–Upper Sorbian data) for training child models. A table of data sizes and how they were used is shown in Appendix A.

4 Preprocessing

We build BPE vocabularies of size 2k, 5k, 10k, 15k, and 20k using `subword-nmt`[4] (Sennrich et al., 2016b). After building the vocabulary, we add a set of 25 generic tags, plus a special backtranslation tag "<BT>", which we use in future experiments for indicating when training data has been backtranslated (Caswell et al., 2019). We also add all Moses and Sockeye special tags (ampersand, <unk> etc.) to a glossary file used for applying BPE, which prevents them from being segmented.

Because there is so much more Czech data than Upper Sorbian data, we duplicate the in-domain parallel hsb-de data and the monolingual HSB data 25 times when training BPE in order to make sure that HSB data is adequately represented (and not overwhelmed by Czech data) in training the encoding. After training BPE, we extract (and fix for the remainder of our experiments) a single DE vocabulary and a single HSB-CS vocabulary, covering all the relevant data used to train BPE for that language pair.

We ran BPE-dropout with a rate of 0.1 over the training data 5 times using the same BPE merge operations, vocabularies and glossaries as before, concatenating these variants to form an extended training set.

		Child Dropout		
		None	Source	Both
Parent Dropout	None	54.6	54.5	54.3
	Source	55.0	**55.5**	54.2
	Both	54.9	**55.5**	55.0

Table 2: Comparison of BPE-dropout use in both parent and child systems for 10k vocabulary DE-HSB translation (measured on devel_test set), without backtranslation. All parent systems were trained on the German-Czech data, while child systems trained on the parallel DE-HSB data. *None* involves no BPE-dropout, *source* applies BPE-dropout to the source side only, and *both* applies it to both the source and the target.

5 Software and Systems

We used Sockeye's (Hieber et al., 2018) implementation of Transformer (Vaswani et al., 2017) with 6 layers, 8 attention heads, network size of 512 units, and feedforward size of 2048 units. We have changed the default gradient clipping type to absolute, used the whole validation set during validation, an initial learning rate of 0.0001, batches of ∼8192 tokens/words, maximum sentence length of 200 tokens, optimizing for BLEU. Parent systems used checkpoint intervals of 2500 and 4000. Child system checkpoint intervals varied from 65 to 4000 to balance frequent checkpointing with efficiency. Decoding was performed with beam size 5.

6 Results and Discussion

6.1 BPE-Dropout in Transfer Learning

Provilkov et al. (2020) examine BPE-dropout when building translation systems for individual language pairs. Here we apply it in a transfer learning setting, raising the question of whether BPE-dropout should be applied to the parent system, the child system, or both, as well as the question of using source-side BPE-dropout or both source- and target-side BPE-dropout.

[1] http://www.opensubtitles.com
[2] http://www.statmt.org/wmt20/translation-task.html
[3] http://www.statmt.org/wmt20/unsup_and_very_low_res/
[4] https://github.com/rsennrich/subword-nmt

Our results for this are somewhat mixed, owing in part to the relatively small BLEU gains produced by BPE-dropout (as compared to backtranslation). In Table 2 we show BLEU scores for German–Upper Sorbian translation with a 10k vocabulary and no backtranslation. The most promising systems in that experiment are those with source-side BPE-dropout in the child system, with either both side or source-side dropout in the parent. In the 20k vocabulary DE-HSB setting with second iteration backtranslation, we saw a similar effect, with source BPE-dropout for both parent and child having a BLEU score of 58.4 on devel_test, +1.1 above the second-best system (no BPE-dropout in parent or child). Results in the other translation direction were more ambiguous, leaving room for future analysis of BPE-dropout in the transfer learning setting.

As a result of these experiments, many of the systems we used in our final ensembles were trained with source-side BPE-dropout, though when it appeared promising we also ensembled with systems without BPE-dropout.

6.2 Iterative Backtranslation

We performed two rounds of backtranslation of Upper Sorbian monolingual data and German monolingual data described in Section 2.5. The first round (BT1) used our strongest system without backtranslation, while the second round (BT2) used our strongest system including backtranslated data from the first round. We ran experiments sweeping BPE vocabulary sizes and backtranslated corpora; for German news we experimented 300k and 600k subsets as well as the full 1.2M line random subselection. In all experiments the 60k sentence-pair parallel HSB-DE corpus was replicated a number of times to approximately match the included backtranslated data in number of lines.

The second round of backtranslation of the Upper Sorbian monolingual data improved the BLEU score by 0.7 BLEU points for the best configuration, with the vocabulary size of the best configuration increasing to 20k from 15k. However, the second round of backtranslation of the German monolingual data did not improve the subsequent HSB-DE systems, instead showing a drop of 0.1 BLEU points; our final system (Section 6.5) uses a mix of systems trained using BT1 and BT2. For full details of the systems used for backtranslation, see Appendix B.

System	DE-HSB	HSB-DE
Baseline	44.2	44.1
Base. + BPE-Dr.	44.4	44.7
Base. + BT2	54.9	54.7
Base. + BT2 + BPE-Dr.	56.1	55.0
Child	54.7	53.4
Child + BPE-Dr.	55.5	54.1
Child + BT2	57.7	56.5
Child + BT2 + BPE-Dr.	58.4	56.8
Final Submitted Systems	59.4	58.9

Table 3: Ablation experiments showing performance of baseline systems, BPE-dropout, backtranslation, transfer learning, and their combination. All systems shown here do not use pseudo-Sorbian. DE-HSB systems here have a 20k vocabulary, while HSB-DE have a 10k vocabulary. BLEU score is reported on devel_test set. The final line shows the submitted primary systems and their performance on devel_test.

Generating multiple translation for backtranslation (i.e. multiple source sentences for each authentic target sentence) is known to improve translation quality (Imamura et al., 2018; Imamura and Sumita, 2018); all of the systems we have described here used a single backtranslation per target sentence. After the submission of our final systems, we experimented with backtranslation using *n*-best translations of the monolingual text. In both directions, we found that building student systems using the 10-best backtranslation list generated with sampling from the softmax's top-10 vocabulary (rather than taking the max), but without BPE-dropout, produced improvements of around 0.2-0.8 BLEU.[5] The resulting systems had comparable BLEU scores to the systems trained with single variant backtranslation and BPE-dropout; we leave as future work an examination of the result of combining multiple backtranslations with BPE-dropout.

6.3 Ablation

Here we first discuss the impact of our non-pseudo-Sorbian approaches: BPE-dropout, backtranslation, and transfer learning, showing how each contributed to the final systems used for ensembling.

Table 3 shows ablation experiments for DE-HSB (20k vocabulary) and HSB-DE (10k vocabulary).[6] In the first four lines, we consider training a system without transfer learning, starting from a base-

[5] Authentic bitext was upsampled to keep the ratio identical to our prior experiments.

[6] Smaller vocabulary sizes perform better on the baseline experiments, but the trends remain the same, so we show results for our final vocabulary sizes.

line built using only the parallel Upper Sorbian–German data. Despite the small data size, and perhaps due to the close match between training and test data, this baseline has high BLEU scores on the devel_test set: 44.2 (DE-HSB) and 44.1 (HSB-DE). Adding BPE-dropout to this setting (with 5 runs of the algorithm) results in a modest improvement (+0.2 BLEU for DE-HSB and +0.6 BLEU for DE-HSB). If we instead add backtranslated data (translated in our second iteration of backtranslation), we see a much larger jump of +10.7 and +10.6 BLEU respectively over the baselines; note that this also represents a huge increase in available data for training. Combining the two approaches adds an additional +1.2 and 0.3 BLEU, respectively.

In fact, these systems outperform both a parent-child baseline and a parent-child system with BPE-dropout, highlighting the importance of incorporating additional target-side monolingual data in the low-resource setting. Once we combine backtranslation we see a moderate improvement over the child systems with BPE-dropout (+2.6 and +2.4 BLEU, respectively). Again, combining BPE-dropout and backtranslation still produces more improvement, as does eventual ensembling.

Due to time constraints, we did not run a full ablation study of word, character and combined pseudo-Sorbian. Our initial results (run with an earlier version of character pseudo-Sorbian, and a differently extracted BPE vocabulary) found for the HSB-DE direction that word pseudo-Sorbian slightly outperformed (on the order of 0.5 BLEU) character pseudo-Sorbian for 10k vocabulary, but was comparable for 2k and 5k vocabulary sizes; these results are given in Appendix C. The combination of the two had slightly higher scores across those three vocabulary sizes (ranging from +0.1 to +0.6 BLEU) than either of the two individual approaches, so we used the combination for the remaining experiments.

6.4 Final German–Upper Sorbian System

System	BLEU
1. Child + BT2	57.7
2. Child + Src. BPE-Dr. + BT2	58.4
3. Pseudo-Sorbian + Child + BT2	57.8
4. Pseudo. + Child + Src. BPE-Dr. + BT2	58.2
Ensemble	**59.4**

Table 4: Primary German–Upper Sorbian ensemble submission BLEU score on devel_test, with scores of each of its individual component systems. The system numbers correspond to the list in Section 6.4.

Our final German–Upper Sorbian system is an ensemble of four systems, with vocabulary size of 20k merges. All child models ensembled were trained on second iteration backtranslated monolingual HSB data (all available, filtered) and 12 replications of the de–hsb parallel text, with backtranslation tags.

1. Child without BPE-dropout, de–cs parent without BPE-dropout.
2. Child with source side BPE-dropout, de–cs parent with source side BPE-dropout
3. Child without BPE-dropout, pseudo-hsb–de parent without BPE-dropout.
4. Child with source side BPE-dropout, pseudo-hsb–de parent with source side BPE-dropout

The system scores on devel_test are shown in Table 4. The best scoring individual systems were transfer learning systems with source-side BPE-dropout, with the one using pseudo-Sorbian falling slightly behind the non-pseudo-Sorbian by 0.2 BLEU points. Without BPE-dropout, the best pseudo-Sorbian system shown here outperforms its corresponding non-pseudo-Sorbian system by approximately 0.1 BLEU. On the test set, this system had scores of (as computed by the Matrix submission) 57.3 BLEU-cased, TER (Snover et al., 2006) of 0.3, BEER 2.0 (Stanojević and Sima'an, 2014) of 0.754, and CharacTER (Wang et al., 2016) of 0.255. This was 3.4 BLEU-cased behind the best-scoring system (SJTU-NICT), but within 0.6 BLEU of the second- and third-highest scoring systems (University of Helsinki); it was also tied with the third-highest scoring system (University of Helsinki) in terms of CharactTER.

6.5 Final Upper Sorbian–German System

System	BLEU
1. Child + BPE-Dr. + BT1	57.2
2. Child + BT2	57.1
3. Pseudo. + Child + BT1	57.2
4. Pseudo. + Child + BPE-dr. + BT1	57.1
5. Pseudo. + Child + BT2	57.1
Ensemble	**58.9**

Table 5: Primary Upper Sorbian–German ensemble submission BLEU score on devel_test, with scores of each of its individual component systems. The numbers correspond to the list in Section 6.5.

The final Upper Sorbian–German system is an ensemble of systems with a BPE vocabulary of 10k merges.

1. Child with source side BPE-dropout, 20 times hsb–de data, 1.2M lines of first iteration backtranslated news data; cs–de parent with source side BPE-dropout
2. Child without BPE-dropout, 25 times hsb–de data, news commentary (NC) and 1.2M lines of news second iteration backtranslated;[7] cs-de parent without BPE-dropout
3. Child without BPE-dropout, 25 times hsb–de data, NC and 1.2M lines of news first iteration backtranslated; pseudo-hsb–de parent without BPE-dropout
4. Child with source side BPE-dropout, 25 times hsb–de data, NC and 1.2M lines of news first iteration backtranslated; pseudo-hsb–de parent with source side BPE-dropout
5. Child without BPE-dropout, 20 times hsb–de data and 1.2M lines of second iteration backtranslated news data; pseudo-hsb–de parent without BPE-dropout

Table 5 shows that the five systems combined were very comparable in BLEU scores (57.1 and 57.2), but their ensembled BLEU score showed an improvement of $\geq$1.7 BLEU over each individual score. The final ensemble had a BLEU-cased score of 58.9 on the test data (calculated by the Matrix submission systems), a TER of 0.29, a BEER 2.0 of 0.579, and a CharacTER score of 0.268. This represented a -0.7 BLEU-cased difference off of the best system (University of Helsinki), but only a -0.001 CharactTER difference.

6.6 Discussion

We experimented with a variety of ensembles, and found that our strongest ensembles were those that included both the pseudo-Sorbian systems and those built without pseudo-Sorbian. In initial experiments with Upper Sorbian-German systems, with vocabulary size 5k, we found that adding pseudo-Sorbian systems to ensembles produced improvements even if the pseudo-Sorbian system did not have quite as high of a BLEU score as the systems built without it. For example, combining the top three systems without pseudo-Sorbian (BLEU scores of 57.3, 57.2, and 57.0, respectively) or the top two of those systems resulted in ensemble system BLEU scores of 57.9. Replacing the third-best system with a pseudo-Sorbian system with a BLEU score of 56.6 resulted in an improved ensemble BLEU score of 58.5. Diverse ensembles (e.g., different architectures or runs) are known to outperform less diverse ensembles (e.g., ensembles of checkpoints) for neural machine translation (Farajian et al., 2016; Denkowski and Neubig, 2017; Liu et al., 2018). While diversity of models for ensembling is usually discussed in terms of model architecture or seeding of multiple runs, we could argue that the use of lexically modified training data could constitute another form of model diversity, contributing to a stronger ensembled model.

For baseline systems trained only on the parallel data, smaller vocabulary sizes performed best, as expected (given only 60,000 lines of text, large vocabulary sizes may contain many tokens that are only observed a small number of times). As we added transfer learning, backtranslation, and eventually ensembling, the best systems were those with slightly larger vocabulary sizes. In the Upper Sorbian–German translation direction, some of our best performing systems that did not use pseudo-Sorbian were found with a 5k vocabulary size, while 10k was generally better for the pseudo-Sorbian systems. We tried ensembles with both 5k and 10k that included pseudo-Sorbian and non-pseudo-Sorbian systems, and found the best results with 10k.

7 Conclusions

In this work, we demonstrated that transfer learning, BPE-dropout, and backtranslation all provide improvements for this low-resource setting. Our experiments on lexical modifications, building pseudo-Sorbian text for training parent models, performed approximately on-par with standard transfer learning approaches, and could be trivially combined with BPE-dropout. While the lexical modification approach did not outperform the standard transfer learning setup, we found that it still improved ensembles, possibly due to the increase in system diversity.

Acknowledgments

We thank the reviewers for their comments and suggestions. We thank Yunli Wang, Chi-kiu Lo, Sowmya Vajjala, and Huda Khayrallah for discussion and feedback.

[7] This version of the second iteration backtranslation differs slightly from that used in the remainder of the experiments, in that UNKs (tokens representing unknown words) were not filtered out.

References

Željko Agić and Ivan Vulić. 2019. JW300: A wide-coverage parallel corpus for low-resource languages. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3204–3210, Florence, Italy. Association for Computational Linguistics.

Isaac Caswell, Ciprian Chelba, and David Grangier. 2019. Tagged back-translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 53–63, Florence, Italy. Association for Computational Linguistics.

Michael Denkowski and Graham Neubig. 2017. Stronger baselines for trustable results in neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 18–27, Vancouver. Association for Computational Linguistics.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648, Atlanta, Georgia. Association for Computational Linguistics.

Marzieh Fadaee, Arianna Bisazza, and Christof Monz. 2017. Data augmentation for low-resource neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 567–573, Vancouver, Canada. Association for Computational Linguistics.

M Amin Farajian, Rajen Chatterjee, Costanza Conforti, Shahab Jalalvand, Vevake Balaraman, Mattia A Di Gangi, Duygu Ataman, Marco Turchi, Matteo Negri, and Marcello Federico. 2016. FBK's neural machine translation systems for IWSLT 2016. In *Proceedings of the ninth International Workshop on Spoken Language Translation, USA*.

Felix Hieber, Tobias Domhan, Michael Denkowski, David Vilar, Artem Sokolov, Ann Clifton, and Matt Post. 2018. The sockeye neural machine translation toolkit at AMTA 2018. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Papers)*, pages 200–207, Boston, MA. Association for Machine Translation in the Americas.

Vu Cong Duy Hoang, Philipp Koehn, Gholamreza Haffari, and Trevor Cohn. 2018. Iterative back-translation for neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 18–24, Melbourne, Australia. Association for Computational Linguistics.

Kenji Imamura, Atsushi Fujita, and Eiichiro Sumita. 2018. Enhancement of encoder and attention using target monolingual corpora in neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 55–63, Melbourne, Australia. Association for Computational Linguistics.

Kenji Imamura and Eiichiro Sumita. 2018. NICT self-training approach to neural machine translation at NMT-2018. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 110–115, Melbourne, Australia. Association for Computational Linguistics.

Tom Kocmi and Ondřej Bojar. 2018. Trivial transfer learning for low-resource neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 244–252, Belgium, Brussels. Association for Computational Linguistics.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. *Proceedings of the 10th Machine Translation Summit (MT Summit)*, pages 79–86.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing.

Surafel M Lakew, Aliia Erofeeva, Matteo Negri, Marcello Federico, and Marco Turchi. 2018. Transfer learning in multilingual neural machine translation with dynamic vocabulary. In *International Workshop on Spoken Language Translation*.

Yu-Hsiang Lin, Chian-Yu Chen, Jean Lee, Zirui Li, Yuyan Zhang, Mengzhou Xia, Shruti Rijhwani, Junxian He, Zhisong Zhang, Xuezhe Ma, Antonios Anastasopoulos, Patrick Littell, and Graham Neubig. 2019. Choosing transfer languages for cross-lingual learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3125–3135, Florence, Italy. Association for Computational Linguistics.

Pierre Lison and Jörg Tiedemann. 2016. OpenSubtitles2016: Extracting large parallel corpora from movie and TV subtitles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 923–929, Portorož, Slovenia. European Language Resources Association (ELRA).

Yuchen Liu, Long Zhou, Yining Wang, Yang Zhao, Jiajun Zhang, and Chengqing Zong. 2018. A comparable study on model averaging, ensembling and reranking in nmt. In *Natural Language Processing and Chinese Computing*, pages 299–308, Cham. Springer International Publishing.

G. Luo, Y. Yang, Y. Yuan, Z. Chen, and A. Ainiwaer. 2019. Hierarchical transfer learning architecture for low-resource neural machine translation. *IEEE Access*, 7:154157–154166.

Graham Neubig and Junjie Hu. 2018. Rapid adaptation of neural machine translation to new languages. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 875–880, Brussels, Belgium. Association for Computational Linguistics.

Toan Q. Nguyen and David Chiang. 2017. Transfer learning across low-resource, related languages for neural machine translation. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 296–301, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Xing Niu, Michael Denkowski, and Marine Carpuat. 2018. Bi-directional neural machine translation with synthetic parallel data. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 84–91, Melbourne, Australia. Association for Computational Linguistics.

Ivan Provilkov, Dmitrii Emelianenko, and Elena Voita. 2020. BPE-dropout: Simple and effective subword regularization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1882–1892, Online. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of association for machine translation in the Americas*, volume 200. Cambridge, MA.

Miloš Stanojević and Khalil Sima'an. 2014. Fitting sentence level translation evaluation with many dense features. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 202–206, Doha, Qatar. Association for Computational Linguistics.

Ralf Steinberger, Andreas Eisele, Szymon Klocek, Spyridon Pilos, and Patrick Schlüter. 2012. DGT-TM: A freely available translation memory in 22 languages. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, pages 454–459, Istanbul, Turkey. European Languages Resources Association (ELRA).

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, pages 2214–2218, Istanbul, Turkey. European Languages Resources Association (ELRA).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 5998–6008.

Weiyue Wang, Jan-Thorsten Peter, Hendrik Rosendahl, and Hermann Ney. 2016. CharacTer: Translation edit rate on character level. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 505–510, Berlin, Germany. Association for Computational Linguistics.

Xinyi Wang, Hieu Pham, Zihang Dai, and Graham Neubig. 2018. SwitchOut: an efficient data augmentation algorithm for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 856–861, Brussels, Belgium. Association for Computational Linguistics.

Zhirui Zhang, Shujie Liu, Mu Li, Ming Zhou, and Enhong Chen. 2018. Joint training for neural machine translation models with monolingual data. In *AAAI*, pages 555–562.

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer learning for low-resource neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1568–1575, Austin, Texas. Association for Computational Linguistics.

A Data

Table 6 shows the data sizes, including the size after filtering for the monolingual Upper Sorbian data, as well as how each dataset was used for BPE training and vocabulary extraction, parent training, and/or child training.

B Backtranslation Details

The configurations used to backtranslate the first round were:

- For monolingual Upper Sorbian, the HSB–DE child system with 5k vocabulary size and both source and target side BPE-dropout for both the HSB–DE system and its CS–DE parent (53.4 BLEU on devel_test)
- For monolingual German, the DE–HSB child with 10k vocabulary size and both source and target side BPE-dropout for both the DE–HSB system and its DE–CS parent (55.0 BLEU on devel_test).

The following configurations were used to backtranslate the second round:

- For monolingual Upper Sorbian, the HSB–DE child system with 5k vocabulary size and source side BPE-dropout for both the HSB–DE system and its CS–DE parent; 25 times hsb–de data, DE news commentary and 1.2M lines of DE news backtranslated (57.25 BLEU on devel_test)
- For monolingual German, the DE–HSB system with 15k vocabulary size and source side BPE-dropout for both the DE–HSB system and its DE–CS parent; 12 times hsb–de data, HSB Sorbian Institute, Witaj, and Web data backtranslated (57.7 BLEU on devel_test).

After the second round of backtranslation, the top configurations were:

- For HSB–DE, the 5k vocabulary size child with source side BPE-dropout for both the HSB–DE system and its CS–DE parent; 20 times hsb–de data, 1.2M lines of (second round) backtranslated DE news (57.15 BLEU on devel_test)
- For monolingual German, the 20k vocabulary size child with source side BPE-dropout for both the DE–HSB system and its DE–CS parent; 12 times hsb–de data, backtranslated (second round) HSB Sorbian Institute, Witaj, and Web data (58.4 BLEU on devel_test).

We note that the second round of backtranslating the German monolingual news data into Upper Sorbian did not improve the BLEU score for the subsequent HSB–DE systems, with the best configuration dropping by 0.1 BLEU points. However, the second round of backtranslation of the Upper Sorbian monolingual data did improve the BLEU score by 0.7 BLEU points for the best configuration, with the vocabulary size of the best configuration increasing to 20k from 15k.

C Pseudo-Sorbian Comparisons and Analysis

Table 7 presents the results of our pseudo-Sorbian comparison discussed in Sections 2.4 and 6.3; as mentioned; we find that both word- and character-level modifications are similar at small vocabulary sizes, but that word-level modification outperforms at a higher vocabulary size. However, at all vocabulary sizes a combination of the two improves over either approach on its own.

It should be noted again that these preliminary results are not directly comparable to other results in this paper (having trained on a smaller corpus, lacking the JW300 documents) and are also not technically constrained (as the word list used to create the character-level replacement was from bilingual dictionaries, not the constrained corpora). In our submitted systems, we created a new character-level system using only the constrained corpora.

As pseudo-Sorbian lexical modification creates a new training corpus, this raises questions of how to appropriately create BPE vocabularies, in particular when the character-level version is used. In word-level pseudo-Sorbian, the resulting corpus still only consists of words found in the original Czech and Upper Sorbian corpora, although the resulting n-gram frequencies will differ somewhat because of some Czech words being replaced by Upper Sorbian ones. Character-level pseudo-Sorbian, however, can create words and character-level n-grams that do not appear in the original corpus at all.[8]

[8] In future work, it would probably be beneficial to guide the output of the modification with a character-level language model trained on target-language data, to better avoid the generation of n-grams that are unlikely or unattested in the target language.

Data	Lines	BPE/Voc.	Parent	Child
train.hsb-de.{de,hsb}	60,000	Y ×25	N	Y
sorbian_institute_monolingual.hsb	339,822	Y ×25	N	Y
web_monolingual_filtered.hsb	131,047	Y ×25	N	Y
witaj_monolingual_filtered.hsb	220,564	Y ×25	N	Y
OpenSubtitles.cs-de.{de,cs}	16,378,674	Y	Y	N
DGT.cs-de.{de,cs}	4,853,298	Y	Y	N
JW300.{de,cs}	1,155,056	Y	Y	N
Europarl.cs-de.{de,cs}	568,572	Y	Y	N
News-Commentary.cs-de.{de,cs}	185,127	Y	Y	N
WMT-News.cs-de.{de,cs}	20,567	Y	Y	N
news.2019.de.shuffled.deduped.de	57,622,797	N	N	Y
news-commentary-v15.dedup.de	233,111	N	N	Y

Table 6: Data and how it was used, whether for BPE training and vocabulary extraction, parent model training, or child model training. Note that the monolingual German news.2019 data was subsampled, and the number of lines shown represents the full set from which the subsample was drawn.

Pseudo-Sorbian	BPE 2k	BPE 5k	BPE 10k
Word-level	51.8	52.6	52.6
Character-level	51.9	52.6	52.1
Both	52.4	52.7	**52.8**

Table 7: Comparison of approaches to create Pseudo-Sorbian corpora for pre-training, by word-level or character-level replacement of Czech text, at different vocabulary sizes. All scores represent BLEU scores on dev-test, in the HSB–DE direction.

The systems in Table 7 use system-specific BPE; that is, the BPE operations and vocabulary are constructed for each specific {pseudo-Sorbian, Upper Sorbian} training corpus. However, in the final submitted systems, we used a fixed vocabulary from the original {Czech, Upper Sorbian} corpus, which made it possible to ensemble pseudo-Sorbian systems with our other systems, giving us better results than either type of system alone. We do not know what effect (negative or positive) this may have on the quality of the pseudo-Sorbian-trained systems (since they would be using a BPE vocabulary for a different set of "languages", and thus may be over-segmented).[9] This raises a number of questions about appropriate choices of BPE models, which increases the complexity of ablation studies beyond what we are able to address in the scope of this paper.

Setting aside the complications of various BPE training schemes, we return to the BPE segmentations used in our final systems to analyze whether pseudo-Sorbian and BPE-dropout do indeed achieve their goals of producing more overlap between the pseudo-Sorbian or Czech training data and the Upper Sorbian data. We consider the devel_test portion of the Upper Sorbian data. With a 10k BPE vocabulary, that test set contains 4540 unique subword types. 62.6% of those types (2840) are observed in the baseline Czech parent model training data, and 52.9% of the training tokens are in that set. After applying BPE-dropout to the Czech parent training data, the percentage of observed types increases slightly, to 63.4% (2878), with 58.9% of the training tokens in that set. With the pseudo-Sorbian combined system, however, we see a much bigger increase in type overlap: 89.0% of the Upper Sorbian devel_test types (4041) were observed at least once in the pseudo-Sorbian parent data, making up 70.9% of the training tokens. Increased coverage of Upper Sorbian devel_test subword tokens during parent training means that embeddings for those subword tokens will be updated during parent model training, hopefully in a way that improves their warm start in the Upper Sorbian student training.[10]

[9] Using our final BPE segmentation does result in a slightly higher number of segmentations per token than a BPE model trained directly on the pseudo-Sorbian (combined version) data.

[10] While we could imagine that in some situations, they might end up with inappropriate representations, we expect those to be improved when the tokens are observed in student model training.

CUNI Systems for the Unsupervised and Very Low Resource Translation Task in WMT20

Ivana Kvapilíková **Tom Kocmi** **Ondřej Bojar**

Charles University, Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics
Malostranské náměstí 25, 118 00 Prague, Czech Republic
<surname>@ufal.mff.cuni.cz

Abstract

This paper presents a description of CUNI systems submitted to the WMT20 task on unsupervised and very low-resource supervised machine translation between German and Upper Sorbian. We experimented with training on synthetic data and pre-training on a related language pair. In the fully unsupervised scenario, we achieved 25.5 and 23.7 BLEU translating from and into Upper Sorbian, respectively. Our low-resource systems relied on transfer learning from German–Czech parallel data and achieved 57.4 BLEU and 56.1 BLEU, which is an improvement of 10 BLEU points over the baseline trained only on the available small German–Upper Sorbian parallel corpus.

1 Introduction

An extensive area of the machine translation (MT) research focuses on training translation systems without large parallel data resources (Artetxe et al., 2018b,a, 2019; Lample et al., 2018a,b). The WMT20 translation competition presents a separate task on unsupervised and very low-resource supervised MT.

The organizers prepared a shared task to explore machine translation on a real-life example of a low-resource language pair of German (de) and Upper Sorbian (hsb). There are around 60k authentic parallel sentences available for this language pair which is not sufficient to train a high-quality MT system in a standard supervised way, and calls for unsupervised pre-training (Conneau and Lample, 2019), data augmentation by synthetically produced sentences (Sennrich et al., 2016a) or transfer learning from different language pairs (Zoph et al., 2016a; Kocmi and Bojar, 2018).

The WMT20 shared task is divided into two tracks. In the unsupervised track, the participants are only allowed to use monolingual German and Upper Sorbian corpora to train their systems; the low-resource track permits the usage of auxiliary parallel corpora in other languages as well as a small parallel corpus in German–Upper Sorbian.

We participate in both tracks in both translation directions. Section 2 describes our participation in the unsupervised track and section 3 describes our systems from the low-resource track. Section 4 introduces transfer learning via Czech (cs) into our low-resource system. We conclude the paper in section 5.

2 Unsupervised MT

Unsupervised machine translation is the task of learning to translate without any parallel data resources at training time. Both neural and phrase-based systems were proposed to solve the task (Lample et al., 2018b). In this work, we train several neural systems and compare the effects of different training approaches.

2.1 Methodology

The key concepts of unsupervised NMT include a shared encoder, shared vocabulary and model initialization (pre-training). The training relies only on monolingual corpora and switches between de-noising, where the model learns to reconstruct corrupted sentences, and online back-translation, where the model first translates a batch of sentences and immediately trains itself on the generated sentence pairs, using the standard cross-entropy MT objective (Artetxe et al., 2018b; Lample et al., 2018a).

We use a 6-layer Transformer architecture for our unsupervised NMT models following the approach by Conneau and Lample (2019). Both the encoder and the decoder are shared across languages.

We first pre-train the encoder and the decoder separately on the task of cross-lingual masked language modelling (XLM) using monolingual

Proceedings of the 5th Conference on Machine Translation (WMT), pages 1123–1128

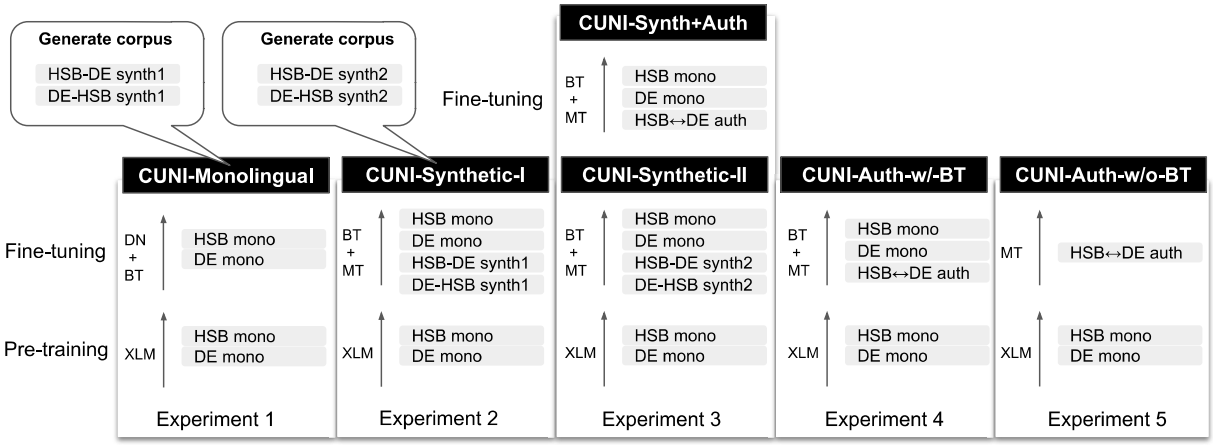


Figure 1: **An overview of selected CUNI systems.** Corpora are illustrated in gray boxes, system names in black boxes. Systems are trained with indicated training objectives: cross-lingual masked language modeling (XLM), denoising (DN), online back-translation (BT), and standard machine translation objective (MT). Monolingual training sets *DE mono* and *HSB mono* were available for both WMT20 task tracks, the parallel training set *HSB↔DE auth* was only allowed in the low-resource supervised track.

data only (Conneau and Lample, 2019). Subsequently, the initialized MT system (*CUNI-Monolingual*) is trained using de-noising and online back-translation. We then use this system to translate our entire monolingual corpus and train a new system (*CUNI-Synthetic-I*) from scratch on the two newly generated synthetic parallel corpora *DE-HSB synth1* and *HSB-DE synth1*. Finally, we use the new system to generate *DE-HSB synth2* and *HSB-DE synth2*, and repeat the training to evaluate the effect of another back-translation round (*CUNI-Synthetic-II*).

All unsupervised systems are trained using the same BPE subword vocabulary (Sennrich et al., 2016b) with 61k items generated using `fastBPE`.[1] An overview of the systems and their training stages is given in fig. 1.

2.2 Data

Our `de` training data comes from News Crawl; the `hsb` data was provided for WMT20 by the Sorbian Institute and the Witaj Sprachzentrum.[2] Most of the `hsb` data was of high quality but we fed the web-scraped corpus (*web_monolingual.hsb*) through a language identification tool `fastText`[3] to identify proper `hsb` sentences. All `de` data was also cleaned using this tool.

The final monolingual training corpora have 22.5M sentences (*DE mono*) and 0.6M sentences (*HSB mono*). Synthetic parallel corpora are generated from the monolingual data sets by coupling the sentences with their translation counterparts as described in section 2.1.

The parallel development (dev) and testing (dev test) data sets of 2k sentence pairs provided by WMT20 organizers are used for parameter tuning and model selection. The final evaluation is run on the blind test set *newstest2020*.

2.3 Results

The resulting scores measured on the blind *newstest2020* are listed in table 1 and table 2. The translation quality metrics BLEU (Papineni et al., 2002), TER (Snover et al., 2006), BEER (Stanojević and Sima'an, 2014) and CharacTER (Wang et al., 2016) provide consistent results. The best quality is reached when using synthetic corpora from the second back-translation iteration, although the second round adds only a slight improvement. A similar observation is made by Hoang et al. (2018) who show that the second round of back-translation does not enhance the system performance as much as the first round. Additionally, the third round does not produce any significant gains.

When training on synthetic parallel corpora, it is still beneficial to perform back-translation on-the-fly (Artetxe et al., 2018b) whereby new training instances of increasing quality are generated in every training step. This method adds 1 - 2 BLEU points to the final score as compared to training

[1] https://github.com/glample/fastBPE
[2] http://www.statmt.org/wmt20/unsup_and_very_low_res/
[3] https://github.com/facebookresearch/fastText/

		newstest2020					*dev test set*
	System Name	**BLEU**	**BLEU-cased**	**TER**	**BEER 2.0**	**CharacTER**	**BLEU**
a	**CUNI-Monolingual**	**23.7**	**23.4**	0.606	**0.530**	0.559	23.4
	CUNI-Synthetic-I	23.4	23.2	0.617	0.531	0.575	22.2
	CUNI-Synthetic-II*	**23.7**	**23.4**	**0.618**	**0.530**	**0.563**	**23.7**
b	**CUNI-Supervised-Baseline**	43.7	43.2	0.439	0.670	0.382	38.7
	CUNI-Auth-w\o-BT	51.6	51.2	0.362	0.710	0.332	48.3
	CUNI-Auth-w\-BT	**54.3**	**53.9**	**0.337**	**0.726**	**0.310**	**52.1**
	CUNI-Synth+Auth*	53.8	53.4	0.343	0.721	0.315	50.5

Table 1: Translation quality of the unsupervised (a) and low-resource supervised (b) `hsb` → `de` systems on *newstest2020* and the unofficial test set. The asterisk * indicates systems submitted into WMT20.

		newstest2020					*dev test set*
	System Name	**BLEU**	**BLEU-cased**	**TER**	**BEER 2.0**	**CharacTER**	**BLEU**
a	**CUNI-Monolingual**	21.7	21.2	0.670	0.497	0.557	20.4
	CUNI-Synthetic-I	24.9	24.5	0.599	0.535	0.521	25.1
	CUNI-Synthetic-II*	**25.5**	**25.0**	**0.592**	**0.540**	**0.516**	**25.3**
b	**CUNI-Supervised-Baseline**	40.8	40.3	0.452	0.655	0.373	38.3
	CUNI-Auth-w\o-BT	47.5	47.1	0.390	0.689	0.336	47.1
	CUNI-Auth-w\-BT	**52.3**	**51.8**	**0.350**	**0.718**	**0.301**	**52.4**
	CUNI-Synth+Auth*	50.6	50.1	0.368	0.703	0.326	50.4

Table 2: Translation quality of the unsupervised (a) and low-resource supervised (b) `de` → `hsb` systems on *newstest2020* and the unofficial test set. The asterisk * indicates systems submitted into WMT20.

only on sentence pairs from the two synthetic corpora so we use it in all our unsupervised systems.

We used the `XLM`[4] toolkit for running the experiments. Language model pre-training took 4 days on 4 GPUs[5]. The translation models were trained on 1 GPU[6] with 8-step gradient accumulation to reach an effective batch size of 8 × 3400 tokens. We used the Adam (Kingma and Ba, 2015) optimizer with inverse square root decay ($\beta_1 = 0.9$, $\beta_2 = 0.98$, $lr = 0.0001$) and greedy decoding.

3 Very Low-Resource Supervised MT

3.1 Methodology

Our systems introduced in this section have the same model architecture as described in section 2, but now we allow the usage of authentic parallel data. We pre-train a bilingual XLM model and fine-tune with either only authentic parallel data (*CUNI-Auth-w\o-BT*) or both parallel and monolingual data, using a combination of standard MT training and online back-translation (*CUNI-Auth-w\-BT*). Finally, we utilize the trained model *CUNI-Synthetic-II* from section 2 and fine-tune it on the authentic parallel corpus, again using standard supervised training as well as online back-translation (*CUNI-Synth+Authentic)*.

All systems are trained with the same BPE subword vocabulary of 61k items.

3.2 Data

In addition to the data described in section 2.2, we used the authentic parallel corpus of 60k sentence pairs provided by Witaj Sprachzentrum mostly from the legal and general domain.

3.3 Results

The resulting scores are listed in the second part of table 1 and table 2. We compare system performance against a supervised baseline which is a vanilla NMT model trained only on the small parallel train set of 60k sentences, without any pre-training or data augmentation.

Our best system gains 11.5 BLEU over this baseline, utilizing the larger monolingual corpora for XLM pre-training and online back-translation. Fine-tuning one of the trained unsupervised systems on parallel data leads to a lower gain of ∼10 BLEU points over the baseline.

The translation models were trained on 1 GPU[7] with 8-step gradient accumulation to reach an effective batch size of 8 × 1600 tokens. Other training details are equivalent to section 2.1.

[4] https://github.com/facebookresearch/XLM

[5] GeForce GTX 1080, 11GB of RAM

[6] Quadro P5000, 16GB of RAM

[7] GeForce GTX 1080 Ti, 11GB of RAM

System Name	BLEU	BLEU-cased	TER	BEER 2.0	CharacTER
Helsinki-NLP	60.0	59.6	0.286	0.761	0.267
NRC-CNRC	59.2	58.9	0.290	0.759	0.268
SJTU-NICT	58.9	58.5	0.296	0.754	0.274
CUNI-Transfer	57.4	56.9	0.307	0.746	0.285
Bilingual only	47.8	47.4	0.394	0.695	0.356

Table 3: Translation quality of `hsb` → `de` systems on newstest2020.

System Name	BLEU	BLEU-cased	TER	BEER 2.0	CharacTER
SJTU-NICT	61.1	60.7	0.283	0.759	0.250
Helsinki-NLP	58.4	57.9	0.297	0.755	0.255
NRC-CNRC	57.7	57.3	0.300	0.754	0.255
CUNI-Transfer	56.1	55.5	0.315	0.743	0.265
Bilingual only	46.8	46.4	0.389	0.692	0.335

Table 4: Translation quality of `de` → `hsb` systems on newstest2020.

4 Very Low-Resource Supervised MT with Transfer Learning

One of the main approaches to improving performance under low-resource conditions is transferring knowledge from different high-resource language pairs (Zoph et al., 2016b; Kocmi and Bojar, 2018). This section describes the unmodified strategy for transfer learning as presented by Kocmi and Bojar (2018), using German–Czech as the parent language pair. Since we do not modify the approach nor tune hyperparameters of the NMT model, we consider our system a transfer learning baseline for low-resource supervised machine translation.

4.1 Methodology

Kocmi and Bojar (2018) proposed an approach to fine-tune a low-resource language pair (called "child") from a pre-trained high-resource language pair (called "parent") model. The method has only one restriction and that is a shared subword vocabulary generated from the corpora of both the child and the parent. The training procedure is as follows: first train an NMT model on the parent parallel corpus until it converges, then replace the training data with the child corpus.

We use the Tensor2Tensor framework (Vaswani et al., 2018) for our transfer learning baseline and model parameters "Transformer-big" as described in (Vaswani et al., 2018). Our shared vocabulary has 32k wordpiece tokens. We use the Adafactor (Shazeer and Stern, 2018) optimizer and a reverse square root decay with 16 000 warm-up steps. For the inference, we use beam search of size 8 and alpha 0.8.

4.2 Data

In addition to the data described in section 3.2, we used the `cs-de` parallel corpora available at the OPUS[8] website: OpenSubtitles, MultiParaCrawl, Europarl, EUBookshop, DGT, EMEA and JRC. The `cs-de` corpus has 21.4M sentence pairs after cleaning with the `fastText` language identification tool.

4.3 Results

We compare the results of our transfer learning baseline called *CUNI-Transfer* with three top performing systems of WMT20. These systems use state-of-the-art techniques such as BPE-dropout, ensembling of models, cross-lingual language modelling, filtering of training data and hyperparameter tuning. Additionally, we also include results for a system we trained without any modifications solely on bilingual parallel data (*Bilingual only*).[9]

The results in table 4 show that training solely on German–Upper Sorbian parallel data leads to a performance of 47.8 BLEU for `de`→`hsb` and 46.7 BLEU for `hsb`→`de`. When using transfer learning with a Czech–German parent, the performance increases by roughly 10 BLEU points to 57.4 and 56.1 BLEU. As demonstrated by the winning system, the performance can be further boosted using additional techniques and approaches to 60.0 and 61.1 BLEU. This shows that transfer learning plays an important role in the low-resource scenario.

[8] `http://opus.nlpl.eu/`

[9] The model *Bilingual only* is trained on the same data as *CUNI-Supervised-Baseline* but uses a different architecture and decoding parameters.

5 Conclusion

We participated in the unsupervised and low-resource supervised translation task of WMT20.

In the fully unsupervised scenario, the best scores of 25.5 (hsb→de) and 23.7 (de→hsb) were achieved using cross-lingual language model pre-training (XLM) and training on synthetic data produced by NMT models from earlier two iterations. We submitted this system under the name *CUNI-Synthetic-II*.

In the low-resource supervised scenario, the best scores of 57.4 (hsb→de) and 56.1 (de→hsb) were achieved by pre-training on a large German–Czech parallel corpus and fine-tuning on the available German–Upper Sorbian parallel corpus. We submitted this system under the name *CUNI-Transfer*.

We showed that transfer learning plays an important role in the low-resource scenario, bringing an improvement of ∼10 BLEU points over a vanilla supervised MT model trained on the small parallel data only. Additional techniques used by other competing teams yield further improvements of up to 4 BLEU over our transfer learning baseline.

Acknowledgments

This study was supported in parts by the grants 19-26934X and 18-24210S of the Czech Science Foundation, SVV 260 575 and GAUK 1050119 of the Charles University. This work has been using language resources and tools stored and distributed by the LINDAT/CLARIN project of the Ministry of Education, Youth and Sports of the Czech Republic (LM2018101).

References

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018a. Unsupervised statistical machine translation. In *Proceedings of the 2018 Conference on EMNLP*, Brussels. Association for Computational Linguistics.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2019. An effective approach to unsupervised machine translation. In *Proceedings of the 57th Annual Meeting of the ACL*, Florence. Association for Computational Linguistics.

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. 2018b. Unsupervised neural machine translation. In *Proceedings of the Sixth International Conference on Learning Representations*.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d' Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 7059–7069. Curran Associates, Inc.

Vu Cong Duy Hoang, Philipp Koehn, Gholamreza Haffari, and Trevor Cohn. 2018. Iterative back-translation for neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference for Learning Representations*.

Tom Kocmi and Ondřej Bojar. 2018. Trivial transfer learning for low-resource neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, Brussels. Association for Computational Linguistics.

Guillaume Lample, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018a. Unsupervised machine translation using monolingual corpora only. In *6th International Conference on Learning Representations (ICLR 2018)*.

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018b. Phrase-based & neural unsupervised machine translation. In *Proceedings of the 2018 Conference on EMNLP*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the ACL*, Philadelphia. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the ACL (Volume 1: Long Papers)*, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the ACL*, Berlin. Association for Computational Linguistics.

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th of the AMTA*, Cambridge. Association for Machine Translation in the Americas.

Miloš Stanojević and Khalil Sima'an. 2014. Fitting sentence level translation evaluation with many dense features. In *Proceedings of the 2014 Conference on EMNLP*, Doha, Qatar. Association for Computational Linguistics.

Ashish Vaswani, Samy Bengio, Eugene Brevdo, Francois Chollet, Aidan Gomez, Stephan Gouws, Llion Jones, Lukasz Kaiser, Nal Kalchbrenner, Niki Parmar, Ryan Sepassi, Noam Shazeer, and Jakob Uszkoreit. 2018. Tensor2tensor for neural machine translation. In *Proceedings of the 13th Conference of the AMTA (Volume 1: Research Papers)*, Boston, MA. Association for Machine Translation in the Americas.

Weiyue Wang, Jan-Thorsten Peter, Hendrik Rosendahl, and Hermann Ney. 2016. CharacTer: Translation edit rate on character level. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, Berlin, Germany. Association for Computational Linguistics.

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016a. Transfer learning for low-resource neural machine translation. In *Proceedings of the 2016 Conference on EMNLP*, Austin, Texas. Association for Computational Linguistics.

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016b. Transfer learning for low-resource neural machine translation. In *Proceedings of the 2016 Conference on EMNLP*, Austin, Texas. Association for Computational Linguistics.

The University of Helsinki and Aalto University submissions to the WMT 2020 news and low-resource translation tasks

Yves Scherrer
University of Helsinki
yves.scherrer@helsinki.fi

Stig-Arne Grönroos
Aalto University
stig-arne.gronroos@aalto.fi

Sami Virpioja
University of Helsinki
sami.virpioja@helsinki.fi

Abstract

This paper describes the joint participation of University of Helsinki and Aalto University to two shared tasks of WMT 2020: the news translation between Inuktitut and English and the low-resource translation between German and Upper Sorbian. For both tasks, our efforts concentrate on efficient use of monolingual and related bilingual corpora with scheduled multi-task learning as well as an optimized subword segmentation with sampling.

Our submission obtained the highest score for Upper Sorbian → German and was ranked second for German → Upper Sorbian according to BLEU scores. For English–Inuktitut, we reached ranks 8 and 10 out of 11 according to BLEU scores.

1 Introduction

Our work is motivated by Grönroos et al. (2020), who provide a detailed study of different transfer learning and regularization approaches for low-resource machine translation. They focus on an asymmetric-resource scenario in which the target language is underresourced, but related to a higher-resource language that can be used in a multilingual setting. For example, in the English-to-Estonian task, Estonian is assumed to be a low-resource language (LRL) which is complemented by a second higher-resource target language (HRL), Finnish. Among the WMT 2020 shared tasks, the **German → Upper Sorbian** low-resource translation task exactly corresponds to this setup, with Czech being a high-resource language closely related to Upper Sorbian. We adapt the approach proposed by Grönroos et al. (2020) also to three slightly different scenarios: in the **Upper Sorbian → German** task, the low-resource language is on the source side, but can be complemented with Czech in the same way; for the **English → Inuktitut** task, no related high-resource language is available; and for **Inuktitut → English**, the low-resource language is on the source side and no high-resource language is available.

Grönroos et al. (2020) recommend the combination of the following techniques to reach optimal translation performance in their examined setup:

Scheduled multi-task learning The learning process is split in two phases. The first phase only sees data from the source and the HRL, whereas LRL data is only added in the second phase.

Backtranslation The addition of synthetic data has become a staple of neural machine translation. They recommend marking synthetic data and controlling its weight in the task scheduler.

Subword regularization Following Kudo (2018), each time a word is used during training, a new segmentation into subwords is sampled from the probabilistic segmentation model.

Monolingual tasks In order to benefit from more easily available monolingual data and to make the model more robust to noise, they propose to include denoising sequence autoencoder tasks. A first variant applies small changes to the input side of the corpus (e.g. word deletions, substitutions and reorderings). A second variant, called taboo sampling, relies on the subword regularization idea and generates two maximally different segmentations of the source and target text. For English–Inuktitut,[1] we extend this idea to a transliteration task between romanized and syllabic Inuktitut.

Subword regularization and taboo sampling require the subword segmentation to be based on

[1] We use dashes to refer to language pairs independently of translation direction.

Proceedings of the 5th Conference on Machine Translation (WMT), pages 1129–1138

	Parallel		Monolingual		
Corpus	EN→IU	IU→EN	EN	IU	IU Translit.
NH train	771 382	771 382			771 382
Wikititles	455	455			455
NH unaligned (EN)		*319 045*			
NH unaligned (IU)	*356 005*			356 005	
NewsCommentary		*557 628*			
NewsCrawl 2019	*2 000 000*		1 000 000		
NewsDiscuss 2019	*2 000 000*		1 000 000		
CommonCrawl	*80 244*			80 244	
Total	1 208 086	5 648 510	2 000 000	436 249	771 837

Table 1: Training corpora sizes (number of lines) for the English–Inuktitut systems. Numbers in italics designate synthetic datasets whose source side is produced by backtranslation.

a probabilistic model. While subword regularization has been introduced in conjunction with SentencePiece, Grönroos et al. (2020) show that the EM+Prune variant of Morfessor (Grönroos et al., 2020) outperforms SentencePiece.

The paper is structured as follows. In Section 2, we present the datasets, their sizes and their usage in our submission. Section 3 reports additional experiments with different approaches to word segmentation. Section 4 provides more details about our multi-task approach and the underlying NMT architecture. Section 5 summarizes the results.

2 Data

Both the Inuktitut–English and Upper Sorbian–German tasks can be qualified as low-resource settings, with less than 800K (deduplicated) parallel training instances for the former and 60K for the latter. For both tasks, we follow the constrained setting, which limits the allowed data to those made available on the WMT website. In this section, we present the parallel and monolingual resources that we used for our systems.

2.1 Inuktitut–English

Training data The training resources for the Inuktitut–English tasks are summarized in Table 1. Two allowed parallel resources are provided, the training part of the Nunavut Hansard (NH) corpus (Joanis et al., 2020) and the small WikiTitles corpus. Since the NH training corpus contained a significant proportion of duplicates and preliminary experiments suggested a slight adverse effect of duplicates, we removed them with the OpusFilter tools (Aulamo et al., 2020). We also cleaned the WikiTitles corpus, removing Inuktitut entries not in syllabic script and identical entries. The Inuktitut side of both training corpora was also used to create a parallel corpus for the romanized ↔ syllabic transliteration task. The romanized version was converted from the syllabic one using the *uniconv* + *iconv* pipeline proposed by the corpus providers.

The NH corpus contains a large amount of unaligned data, which we used as additional monolingual corpora. We removed all sentences that were already covered by one of the parallel NH datasets. The English and Inuktitut parts were processed separately. Both parts were backtranslated to the other language using baseline models trained on the parallel corpora, and filters were applied to both sides of the parallel datasets (see below). The Inuktitut unaligned data was used both as a monolingual dataset and as a synthetic parallel dataset for the EN→IU task, whereas the English unaligned data was only used as a synthetic parallel dataset for the IU→EN task (see Table 1).

Among the wealth of monolingual English data provided by WMT, we selected the NewsCommentary corpus and the 2019 sections of NewsCrawl and NewsDiscuss. We produced Inuktitut backtranslations for NewsCommentary and for 2M sentences each (after filtering) of the NewsCrawl and NewsDiscuss corpora. Of the latter two corpora, we held out distinct sets of 1M sentences each for monolingual tasks.

In terms of monolingual Inuktitut data, besides the unaligned NH data, the organizers only provided a CommonCrawl dump. This corpus was again backtranslated to English and filtered. The resulting corpus was used both as a monolingual

	Parallel				Monolingual		
Corpus	DE→HSB	HSB→DE	DE↔CS	HSB→CS	DE	HSB	CS
Training	60 000	60 000					
Europarl		*560 608*	567 422	*568 573*			
JW300		*1 114 024*	1 140 474	*1 161 656*			
NewsComm.			184 341	*185 132*			
Tatoeba		*4 425*	4 431	*4 448*			
Sorb. Inst.	*334 643*					334 643	
Sorb. Web	*94 980*					94 980	
Witaj	*218 249*					218 249	
NewsComm. (mono)		*389 199*			389 199		184 341
NewsCrawl 2018		*11 529 295*			11 529 295		6 723 691
NewsCrawl 2019		*9 041 245*			9 041 245		9 508 788
Total	707 872	22 698 796	1 896 668	1 919 809	20 959 739	647 872	16 416 820

Table 2: Training corpora sizes (number of lines) for the German–Sorbian systems. Numbers in italics designate synthetic datasets whose source side is produced by backtranslation.

dataset and as a synthetic parallel dataset for the EN→IU task.

Validation data We used the NH *dev* partition as primary validation set, and the *devtest*, *test* and *NewsDev2020* as secondary validation sets.

Preprocessing All datasets were processed with a translation-direction-specific pipeline. Inuktitut spelling and apostrophe normalization scripts were applied both on source and target sides. The Moses punctuation normalization script was applied only to the English target sides of the parallel corpora. No further preprocessing or tokenization was applied.

Filtering The monolingual and backtranslated parallel corpora were filtered with OpusFilter (Aulamo et al., 2020). The main purpose of this step was to remove too short (i.e., less than 1 word or less than 5 characters on either side) and too long sentences (i.e., more than 300 words or 3000 characters on either side). Furthermore, since crawled input data could be noisy and backtranslation could produce suboptimal results for certain sentences, we applied an additional language model filter based on 5-gram language models trained on the NH training part. Sentences with an average character cross-entropy higher than 30 on either side were removed.

2.2 Upper Sorbian–German

Training data The training data for the Upper Sorbian–German tasks are summarized in Table 2. The organizers provide a parallel German–Sorbian corpus of 60k sentence pairs that we use without further filtering or processing. Moreover, we use four sources of parallel German–Czech data for both directions: the Europarl and JW300 corpora provided on OPUS, as suggested by the organizers, and additionally the Tatoeba and NewsCommentary corpora, which are also available through OPUS (Tiedemann, 2012). The German side of three datasets[2] is backtranslated to Upper Sorbian using a baseline system. The Czech side of the four datasets is backtranslated to Upper Sorbian using an unsupervised character-level translation system (see below). Length filters are applied to all data from external resources (see below).

The organizers provide three monolingual Sorbian corpora: *Sorbian Institute*, a Sorbian *Web Crawl*, and *Witaj*. All corpora are backtranslated to German using a baseline system and filtered.

As monolingual German and Czech resources, we selected the NewsCommentary corpus and the 2018 and 2019 sections of NewsCrawl. These datasets were again filtered. The German datasets were backtranslated to Sorbian.

Validation data We use the *dev* partition as primary validation data and the *devtest* partition as secondary validation data (2000 sentence pairs each).[3]

[2]The full (i.e., unaligned) German version of NewsCommentary is also backtranslated, see below.

[3]The validation and test data for Sorbian consist of fairly short and syntactically simple sentences, which explains why even baseline systems such as those reported in Table 4 obtain BLEU scores around 50.

	Segmentation model and parameters	EN→IU BLEU			IU→EN BLEU		
		Dev	Devtest	Test	Dev	Devtest	Test
0	BPE, raw data, 2k+2k/5k+5k separate, no sampling	24.2	17.9	19.3	41.4	31.4	35.0
1	SentencePiece, raw data, 20k+20k separate, no sampling	23.1	16.9	18.4	36.8	27.2	30.9
2	SentencePiece, raw data, 5k+5k separate, no sampling	24.3	18.0	19.3	40.7	30.9	34.3
3	SentencePiece, raw data, 10k joint, no sampling	24.1	18.0	19.5	40.8	30.8	34.3
4	SentencePiece, dedup data, 10k joint, no sampling	24.2	17.7	19.0	40.7	30.8	34.4
5	SentencePiece, dedup data, 10k joint, with sampling	24.0	17.8	19.2	40.6	30.7	34.4
6	Morfessor, dedup data, 10k joint, no sampling	24.1	17.6	19.0	40.5	30.2	33.9
7	Morfessor, dedup data, 10k joint, with sampling	24.4	18.1	19.3	40.5	30.5	34.2

Table 3: Segmentation model experiments for English–Inuktitut. The baseline model (0) was trained using a Sockeye Transformer with default settings, whereas models 1–7 were trained using OpenNMT-py Transformers with default settings. The segmentation models were trained on the raw or deduplicated versions of the NH training corpus.

For the training phases using exclusively German and Czech data, we use the aligned WMT-News corpus (20 549 sentence pairs), made available on OPUS, as validation set.

Filtering A simple length filter was applied to all corpora sourced from OPUS: sentence pairs where at least one side is empty or longer than 300 words were removed. The same filter was also applied to parallel corpora obtained by backtranslation, which explains the slightly diverging numbers for identical corpora in Table 2.

The Sorbian web crawl was filtered by a 5-gram language model trained on the remaining original Sorbian data. Sentences with a cross-entropy higher than 50 were removed.

All corpus filtering tasks were implemented with OpusFilter (Aulamo et al., 2020). No other preprocessing or tokenization was applied.

Czech–Sorbian backtranslation The task organizers do not provide any Czech–Sorbian parallel corpora that could be used to train a baseline system for producing backtranslations. We therefore resort to unsupervised machine translation. Since Czech and Sorbian are closely related, we extract word n-grams from monolingual corpora and match them using string similarity and frequency criteria.[4] This results in a list of 620k distinct bigram pairs and 230k distinct trigram pairs. They are weighted by frequency to constitute a training corpus for a character-level Czech-to-Sorbian translation system. The translation system is based on bi-directional RNNs with two encoder and two decoder layers. In order to produce backtranslations, the Czech input sentences are chunked into overlapping trigram sequences, translated to Sorbian and merged back again.

[4]We use Europarl, NewsCommentary, Taoeba and WMT-News as Czech monolingual corpora, and Training, Sorbian Institute and Witaj as Sorbian monolingual corpora.

3 Segmentation models

NMT models should ideally be able to represent the entire vocabulary of their source and target languages. The simplest solution however, in which word forms are represented as atomic vocabulary items, leads to sparse statistics, issues with out-of-vocabulary words, and heavy computational costs due to large vocabularies. Moreover, such word-level modeling does not allow the productive recombination of morphemes and is thus unsuitable for morphologically rich languages such as Inuktitut or Sorbian. In recent years, a consensus has emerged that NMT vocabularies should consist of subwords of variable size. Various unsupervised word segmentation algorithms have been proposed, among which byte-pair encoding (BPE) (Sennrich et al., 2016), SentencePiece (Kudo and Richardson, 2018), and several variants of Morfessor (Ataman et al., 2017; Banerjee and Bhattacharyya, 2018; Grönroos et al., 2018, 2020).

Besides the actual word segmentation algorithm, various parameters influence the quality of the resulting translation system:

- Separate word segmentation models for each language or one joint vocabulary for all languages. The joint approach scales better to multilingual models, and enables consistent segmentation of named entities and cognate

		Segmentation model	Translation model	DE→HSB BLEU		HSB→DE BLEU	
	Algorithm	Training data (tokens)	Training data (lines)	Dev	Devtest	Dev	Devtest
1	SentencePiece	0.6M HSB + 0.7M DE	60k	56.93	49.76	57.11	48.74
2	Morfessor	0.6M HSB + 0.7M DE	60k	53.42	46.93	53.93	45.79
3	SentencePiece	8.4M HSB + 8.9M DE	60k	57.39	51.00	57.69	49.91
4	Morfessor	8.4M HSB + 8.9M DE	60k	55.34	48.99	55.51	47.61
5	SentencePiece	8.4M HSB + 8.9M DE + 8.4M CS	60k	57.82	51.30	58.45	49.86
6	Morfessor	8.4M HSB + 8.9M DE + 8.4M CS	60k	56.27	49.76	56.68	48.81
7	SentencePiece	8.4M HSB + 8.9M DE + 8.4M CS	708k / 1931k	61.90	55.06	62.41	53.78
8	Morfessor	8.4M HSB + 8.9M DE + 8.4M CS	708k / 1931k	61.56	55.04	62.16	53.83

Table 4: Segmentation model experiments for German–Upper Sorbian. All segmentation models are joint models with 20 000 units, but trained on variable amounts of data. All translation models are OpenNMT-py Transformers with default settings with active subword sampling, trained either without (1–6) or with (7–8) additional backtranslations.

words across languages, assuming they are written in the same script.

- The chosen vocabulary size and the amount of training data from which the segmentation model is learned. Denkowski and Neubig (2017) recommend a vocabulary size of 32k units, trained jointly on all languages, for normal-sized datasets. In contrast, Ding et al. (2019) obtain the best results with small vocabularies of only 500 units in low-resource scenarios. Optimal vocabulary size varies thus depending on the size of the parallel and monolingual data.

- If the segmentation algorithm is based on a probabilistic model (such as SentencePiece or Morfessor), it can be used to sample different segmentations for any given word. This technique is known as subword regularization (Kudo, 2018) and has been shown to improve the robustness of translation models.

Grönroos et al. (2020) tested various segmentation model configurations on a multilingual translation task and obtained best results with Morfessor EM+Prune, followed by SentencePiece and BPE. Furthermore, when trained on the same amount of data and using subword regularization, the vocabulary size (tested between 5K and 20K entries) turned out to be irrelevant for both SentencePiece and Morfessor EM+Prune.

We carried out some additional experiments with the English–Inuktitut task, which differs from their setup in the sense that the languages use different scripts and there is no third language involved. Table 3 compares different parameter settings with the baseline results provided by the organizers (Joanis et al., 2020). A first set of experiments shows that the vocabulary size does matter when not using subword sampling (1 vs 2), but that separate and joint segmentation models perform equivalently (2 vs 3). SentencePiece does not perform better than BPE (2 vs 0), although different preprocessing choices may be responsible for the generally lower results obtained in the IU→EN direction. The second set of experiments shows that Morfessor EM+Prune lags slightly behind SentencePiece when not using sampling (6 vs 4), but that sampling has a more beneficial effect to Morfessor EM+Prune than to SentencePiece (7 vs 6, 5 vs 4).

For German–Upper Sorbian, the setup differs from Grönroos et al. (2020) with respect to the amount of available training data. We therefore ran additional experiments to measure the impact of both the training data used for the segmentation model and the training data used for the translation model. Table 4 summarizes our findings. All experiments are based on joint word segmentation models with a total of 20K vocabulary items.

When training both the segmentation model and the translation model on the provided parallel data (experiments 1 and 2), SentencePiece performs much better than Morfessor EM+Prune. The addition of monolingual training data for the segmentation model (experiments 3 and 4) helps both segmentation algorithms about equally well (+ 1–2 BLEU). In contrast, the further addition of Czech data for the segmentation model (experiments 5 and 6) benefits Morfessor more than SentencePiece on average.[5] Finally, augmenting the trans-

[5]The additional monolingual Sorbian data comes from the

	Training data	Weighting	Monoling. tasks	EN→IU BLEU NH Dev	EN→IU BLEU Newsdev	IU→EN BLEU NH Dev	IU→EN BLEU Newsdev
1	EN↔IU + BT	—	—	24.13	15.72	41.07	32.86
2	EN↔IU + BT	✓	Noise + Translit.	*25.15	*15.95	**41.89**	33.47
3	EN↔IU + BT	✓	Noise + Taboo	**25.28**	**16.15**	*41.67	***33.49**

Table 5: Inuktitut translation experiments. Systems marked with * were used for the final primary submissions.

lation model training data with backtranslations obviously increases the overall translation scores, but also brings Morfessor EM+Prune on par with SentencePiece.

We were thus not able to reproduce the substantial gains in translation quality with Morfessor EM+Prune observed by Grönroos et al. (2020). Rather, we found that SentencePiece was generally more robust to different data conditions and setups. Nevertheless, Morfessor EM+Prune remains competitive with its default parameters if subword sampling is enabled and the training data are carefully chosen. For the final Inuktitut models, we decided to used configuration 7 from Table 3, since it allowed us to use monolingual tasks relying on subword sampling. For the final Sorbian models, we used configuration 8 from Table 4.

4 Translation models

All our models are based on the Transformer architecture and use, by and large, the same hyperparameters as Grönroos et al. (2020). The Transformer contains 8 encoder and 8 decoder layers with 16 attention heads each. The hidden layer size is 1024, the filter size 4096. The minibatch varies between 7200 and 9200 tokens, depending on the task, and gradients are accumulated over 4 minibatches. All models were trained for 200 000 steps, which corresponded to 5–7 days training time on a single V100 GPU. The best savepoint was selected on the basis of development set accuracy; this measure turned out to be more stable than development set BLEU score.

We use the *dynamicdata* branch of the OpenNMT-py toolkit (Klein et al., 2017) for our experiments.[6] This branch provides the necessary adaptations for the techniques introduced by Grönroos et al. (2020): scheduled multi-task learning requires the ability to adjust the task mix during training, whereas subword regularization and the denoising sentence autoencoder task require sampling fresh noise for each minibatch.

The experiments presented in Tables 3 and 4 already confirmed the positive impact of subword regularization and backtranslation. Row 1 of Tables 5 and 6 provide baseline results with these two techniques. Backtranslated training instances are marked with a special token.

Scheduled multi-task learning As row 2 in Table 6 shows, the mere inclusion of a German↔Czech task with language labels but without any task scheduling already increases BLEU scores by 1.5 points. However, simple transfer learning setups such as this are prone to catastrophic forgetting, especially in low-resource settings such as ours.

Kiperwasser and Ballesteros (2018) propose a general strategy called scheduled multi-task learning, in which different tasks are mixed according to a task-mix schedule. Grönroos et al. (2020) propose a partwise constant task-mix schedule with an arbitrary number of steps, any of which can be mixing multiple tasks. This flexibility is useful when training with a large number of heterogeneous tasks: multiple language pairs with different amounts of data, data from different domains (oversampling the in-domain data), natural vs synthetic (e.g. back-translated) data, and auxiliary tasks (e.g. autoencoder).

A training schedule with two phases (row 3 in Table 6) further increases scores slightly. Details of the schedule and the task weights are given in Table 7.

Witaj and Sorbian Institute corpora. We added an equivalent amount of German data from NewsCommentary and WMT-News. The Czech data also stems from NewsCommentary and WMT-News and is complemented by a subset of Czech Europarl.

[6] https://github.com/Waino/OpenNMT-py
The functionality of the *dynamicdata* branch is included by default in the upcoming release v2.0 of OpenNMT-py, albeit in a different implementation.

	Training data	Weight./Schedul.	Monoling. tasks	DE→HSB BLEU Dev	DE→HSB BLEU Devtest	HSB→DE BLEU Dev	HSB→DE BLEU Devtest
1	DE↔HSB + BT	—	—	61.56	55.04	62.16	53.83
2	DE↔CS + DE↔HSB + BT	—	—	63.15	56.71		
3	DE↔CS + DE↔HSB + BT	✓	—	***63.93**	*56.82	64.48	56.27
4	DE↔CS + DE↔HSB + BT	✓	Noise	63.84	56.45	***64.88**	*56.76
5	DE↔CS + DE↔HSB + BT	✓	Taboo	63.61	**57.11**	64.72	**56.96**

Table 6: Sorbian translation experiments. Systems marked with * were used for the final primary submissions.

4.1 Monolingual tasks

Denoising sequence autoencoder task. In the denoising autoencoder (Vincent et al., 2008; Hill et al., 2016) clean text is corrupted by sampling from a noise model, and fed in as a pseudo-source. The target is a reconstruction of the clean input. The goal of the autoencoder tasks is to use monolingual data to strengthen target language modeling in the decoder and source language understanding in the encoder. In addition, the autoencoder task acts as regularization. Noise has been used as a regularizer in many NLP techniques, including dropout (Srivastava et al., 2014), label smoothing (Szegedy et al., 2016), SwitchOut (Wang et al., 2018), and subword regularization (Kudo, 2018). Sampling fresh noise for each minibatch is important, especially in low-resource conditions where the small data set is reused for many epochs. The denoising sequence autoencoder has previously been applied to language model pretraining in BART (Lewis et al., 2019).

Typical noise models for denoising sequence autoencoder apply small changes to the input side of the corpus: local reordering (Lample et al., 2018), deletions (Iyyer et al., 2015), insertions (Vaibhav et al., 2019), substitutions (Wang et al., 2018), and masking (Devlin et al., 2019). Of these, our method applies local reordering and token deletion.

Taboo sampling segmentation task. Grönroos et al. (2020) propose taboo sampling, a noise model extending the subword regularization idea specifically for monolingual data. It takes in monolingual text and generates two maximally different segmentations, e.g. *dys* + *functional* on the source side and *dysfunction* + *al* on the target side. During taboo sampling, all multi-character subwords used in the first segmentation have their probability temporarily set to zero, to ensure that they are not used in the second segmentation.

Transliteration task. As an alternative to taboo sampling, we take advantage of the fact that Inuktitut can be written in two different scripts, romanized and syllabic. Since the segmentation model is trained only on syllabic Inuktitut (and the occasional romanized proper name occurring on the English side of the NH corpus), we assume that the same word will be segmented very differently in the two scripts, leading to a similar effect as taboo sampling. We include a romanized→syllabic transliteration task in the EN→IU model, and a syllabic→romanized task in the IU→EN model.

Experiments 2–3 in Table 5 as well as experiments 4–5 in Table 6 explore different combinations of monolingual tasks. For Inuktitut, the addition of monolingual tasks increases BLEU scores markedly, but there is no clear winner between the transliteration and taboo tasks. For Sorbian, the monolingual tasks only help when translating towards German, but not when translating towards Sorbian. One reason for this somewhat surprising finding could be that the Sorbian monolingual data is identical with the Sorbian target of the backtranslations, so that no additional data is added with the monolingual tasks.

5 Submissions and results

For the best-performing configurations, we trained two models each, one ("basic") with the hyperparameters listed above, and an alternative one with relative position distance clipping at 4 (see Shaw et al., 2018). However, this setting did not yield any consistent accuracy gains or losses.

For the **Inuktitut** task, we submitted single systems of settings 2 and 3 for both directions. For EN→IU, the alternative model of setting 2 obtained the best scores on the test set (10.1 BLEU / 0.301 chrF), whereas for IU→EN, the basic model of setting 3 obtained the best scores on the test set (23.0 BLEU / 0.455 chrF). Among the 11 primary submissions in both translation directions, our sub-

	EN→IU	IU→EN		DE→HSB		HSB→DE	
Training steps	0–200k	0–200k	Training steps	0–60k	60–200k	0–60k	60–200k
Bilingual	45%	45%	Bilingual DE↔CS	90%	50%	85%	25%
Backtranslation	40%	40%	Bilingual DE↔HSB		20%		30%
Noise EN	5%	5%	Backtr. DE↔HSB		20%		30%
Noise IU	5%	5%	Backtr. HSB→CS			5%	5%
Taboo IU / Translit. rom.→syll.	5%		Noise / Taboo DE	5%		5%	5%
Taboo EN / Translit. syll.→rom.		5%	Noise / Taboo CS	5%		5%	
			Noise / Taboo HSB		10%		5%

Table 7: Task schedules for the Inuktitut (left) and Sorbian (right) experiments.

mission obtained rank 8 for EN→IU and rank 10 for IU→EN in terms of (inofficial) BLEU scores. It will be instructive to examine the manual evaluation results and the other system descriptions to identify the reasons behind these rather disappointing results.

For the **Sorbian** task, we submitted ensembles of the basic and alternative models. Setting 3 turned out to be the best choice for DE→HSB (57.9 BLEU / second rank), and setting 4 for HSB→DE (59.6 BLEU / first rank). For both directions, ensembling has raised the BLEU scores by 0.6. Our submissions would obtain first rank in both directions if only single systems were considered.

6 Conclusion

In this work, we tested various methods for low-resource machine translation proposed by Grönroos et al. (2020) on the English–Inuktitut and German–Upper Sorbian tasks in WMT 2020. In particular, we investigated several subword segmentation approaches and the inclusion of monolingual tasks.

In terms of **subword segmentation**, we were not able to reproduce the reported gains for the Morfessor EM+Prune method over SentencePiece. We obtained comparable results with both methods though. We also found that increasing the size of segmentation model training data was useful, and that Morfessor EM+Prune was more sensitive to training data size than SentencePiece. Furthermore, we obtained slight improvements from subword sampling, confirming earlier results.

During the development phase, we also found curious interactions between the subword vocabulary size and different NMT toolkits. We were able to reproduce the organizer-provided Inuktitut baselines with both small and large vocabularies using the Sockeye toolkit, but obtained significantly lower scores with OpenNMT-py and large vocabularies, even after harmonizing the training hyper-parameters between toolkits. With small subword vocabularies, OpenNMT-py became competitive again.

The inclusion of **monolingual tasks** yielded clear improvements for the Inuktitut experiments. The noise model had the most positive effect, whereas the transliteration and taboo sampling tasks showed minor effects. In contrast, the effect of the monolingual tasks on the Sorbian experiments was more subtle. The **two-phase training schedule** introduced by Grönroos et al. (2020) proved useful in the Sorbian experiments.

Acknowledgments

This work is part of the FoTran project, funded by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 771113).

The authors gratefully acknowledge the support of the CSC – IT Center for Science, Finland, for computational resources. We also acknowledge NVIDIA for their GPU grant.

References

Duygu Ataman, Matteo Negri, Marco Turchi, and Marcello Federico. 2017. Linguistically motivated vocabulary reduction for neural machine translation from Turkish to English. *The Prague Bulletin of Mathematical Linguistics*, 108(1):331–342.

Mikko Aulamo, Umut Sulubacak, Sami Virpioja, and Jörg Tiedemann. 2020. OpusTools and parallel corpus diagnostics. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 3782–3789, Marseille, France. European Language Resources Association.

Tamali Banerjee and Pushpak Bhattacharyya. 2018. Meaningless yet meaningful: Morphology grounded subword-level NMT. In *Proceedings of the Second Workshop on Subword/Character Level Models*,

pages 55–60. Association for Computational Linguistics.

Michael Denkowski and Graham Neubig. 2017. Stronger baselines for trustable results in neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 18–27, Vancouver. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional Transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Shuoyang Ding, Adithya Renduchintala, and Kevin Duh. 2019. A call for prudent choice of subword merge operations in neural machine translation. In *Proceedings of Machine Translation Summit XVII Volume 1: Research Track*, pages 204–213, Dublin, Ireland. European Association for Machine Translation.

Stig-Arne Grönroos, Sami Virpioja, and Mikko Kurimo. 2018. Cognate-aware morphological segmentation for multilingual neural translation. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 386–393, Belgium, Brussels. Association for Computational Linguistics.

Stig-Arne Grönroos, Sami Virpioja, and Mikko Kurimo. 2020. Morfessor EM+Prune: Improved subword segmentation with expectation maximization and pruning. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 3944–3953, Marseille, France. European Language Resources Association.

Stig-Arne Grönroos, Sami Virpioja, and Mikko Kurimo. 2020. Transfer learning and subword sampling for asymmetric-resource one-to-many neural translation. ArXiv:2004.04002 [cs.CL].

Felix Hill, Kyunghyun Cho, and Anna Korhonen. 2016. Learning distributed representations of sentences from unlabelled data. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1367–1377.

Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, and Hal Daumé III. 2015. Deep unordered composition rivals syntactic methods for text classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (ACL-IJCNLP) (Volume 1: Long Papers)*, pages 1681–1691.

Eric Joanis, Rebecca Knowles, Roland Kuhn, Samuel Larkin, Patrick Littell, Chi-kiu Lo, Darlene Stewart, and Jeffrey Micher. 2020. The Nunavut hansard Inuktitut–English parallel corpus 3.0 with preliminary machine translation results. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 2562–2572, Marseille, France. European Language Resources Association.

Eliyahu Kiperwasser and Miguel Ballesteros. 2018. Scheduled multi-task learning: From syntax to translation. *Transactions of the Association for Computational Linguistics*, 6:225–240.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Vancouver, Canada. Association for Computational Linguistics.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Unsupervised machine translation using monolingual corpora only. In *International Conference on Learning Representations (ICLR)*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. ArXiv:1910.13461 [cs.CL].

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 464–468, New Orleans, Louisiana. Association for Computational Linguistics.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2214–2218, Istanbul, Turkey. European Language Resources Association (ELRA).

Vaibhav Vaibhav, Sumeet Singh, Craig Stewart, and Graham Neubig. 2019. Improving robustness of machine translation with synthetic noise. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1916–1920.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. 2008. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning (ICML)*, pages 1096–1103.

Xinyi Wang, Hieu Pham, Zihang Dai, and Graham Neubig. 2018. Switchout: an efficient data augmentation algorithm for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 856–861.

The NITS-CNLP System for the Unsupervised MT Task at WMT 2020

Salam Michael Singh **Thoudam Doren Singh** **Sivaji Bandyopadhyay**
Center for Natural Language Processing (CNLP) and Dept. of Computer Science & Engg.
National Institute of Technology Silchar, India
{salammichaelcse,thoudam.doren,sivaji.cse.ju}@gmail.com

Abstract

We describe NITS-CNLP's submission to WMT 2020 unsupervised machine translation shared task for German language (de) to Upper Sorbian (hsb) in a constrained setting i.e, using only the data provided by the organizers. We train our unsupervised model using monolingual data from both the languages by jointly pre-training the encoder and decoder and fine-tune using backtranslation loss. The final model uses the source side (de) monolingual data and the target side (hsb) synthetic data as a pseudo-parallel data to train a pseudo-supervised system which is tuned using the provided development set(dev set).

1 Introduction

This paper provides the system description of the unsupervised neural machine translation system for German to Upper Sorbian submitted by the Center for Natural Language Processing of National Institute of Technology, Silchar, India (NITS-CNLP) in the WMT 2020 shared task for Unsupervised and Very Low Resource machine translation for German and Upper-Sorbian language pair. Specifically, we made our primary submission for the unsupervised task in $de \rightarrow hsb$ direction. We use the data provided by the organisers only i.e, in a constrained manner. Our unsupervised neural machine translation (UNMT) system first pre-trains a transformer (Vaswani et al., 2017) based encoder and decoder model using masked sequence to sequence (MASS) pre-training (Song et al., 2019) and fine-tune using the back-translation (Sennrich et al., 2016a) loss. The final model trained using MASS objective is then used to translate the source side (M_{de}) monolingual data into a synthetic target side data ($M^{'}_{hsb}$)and then train a pseudo-supervised model using $\{M_{de}$,$M^{'}_{hsb}\}$ from scratch.

The remaining of the paper is arranged in following manner: Section 2 gives a brief background of an unsupervised MT. Section 3 describes the data preprocessing. In Section 4, we describe our UNMT system. The results and analysis are shown in Section 5. Finally, Section 6 concludes the paper.

2 Background

NMT (Kalchbrenner and Blunsom, 2013; Cho et al., 2014; Bahdanau et al., 2014) has become the de-facto MT system in recent times achieving near human level translation quality for many language pair however at the cost of millions of bi-text data. Unfortunately, bi-text data for many languages is scarce or non-existent. Unsupervised MT (Lample et al., 2018a; Artetxe et al., 2018b) is one of the techniques to handle the bi-text unavailability by exploiting monolingual data (Sennrich et al., 2016a). Primitive unsupervised MT first maps the monolingual data into a common cross-lingual shared vector embedding space (Conneau et al., 2017; Artetxe et al., 2017) and infer a bilingual dictionary from this shared space using adversarial training (Lample et al., 2018a) or through self learning (Artetxe et al., 2018b) and further improve the model through a combination of de-noising auto-encoder and iterative or on-the-fly back-translation. Subsequently, this principle has been applied in SMT (Lample et al., 2018b; Artetxe et al., 2018a) or a combination of NMT and SMT (Marie and Fujita, 2018; Ren et al., 2019) to further improve the unsupervised MT. However, in this work, we follow a newer approach of cross-lingual language model pretraining (Lample and Conneau, 2019; Song et al., 2019) which has shown to be a stronger initialization for unsupervised MT than the cross-lingual shared vector embedding space.

3 Data and Preprocessing

This section is further divided into two subsections briefing the data description and the preprocessing steps used.

Proceedings of the 5th Conference on Machine Translation (WMT), pages 1139–1143

Corpus		Sentences
mono	de (News Crawl)	5 **M**
	hsb	756.3 **K**
dev/test	de	2 **K**
	hsb	2 **K**

Table 1: Statistics of the monolingual and the dev/test set.

3.1 Data Description

We use a randomly sampled 5**M** monolingual corpus for German side from News Crawl[1] dataset, while we use all the available monolingual data[2] and the parallel side[3] of Upper Sorbian[4] as the combined monolingual data for the same and summing up 756,271 number of sentences. For tuning and evaluation[5], we use the provided devtest[6] data with 2000 sentences for both the dev and test files as shown in Table 1.

3.2 Preprocessing

We use `Moses`(Koehn et al., 2007) toolkit for preprocessing the data. The corpus underwent removal of non-printing characters and tokenization. For the Upper Sorbian, we used Czech `(cs)` language code for tokenization as Upper Sorbian `(hsb)` language code is unavailable in `Moses` toolkit[7] and considering the relatedness of these languages[8].

The above preprocessing is used by MASS pretrain and MASS finetune models while the pseudo-supervised model uses the raw data and learns a Sentencepiece BPE. The details are described in Section 4.2.

4 UNMT System

Our UNMT system is a pipeline of encoder-decoder pretraining and fine-tuning using MASS (Song et al., 2019) and using the synthetic data generated (M'_{hsb}) from the source monolingual data (M_{de}) to train a forward model from scratch. This section is further divided into two subsections, first describing the MASS pretraining and fine-tuning and second, the transformer based forward ($\overrightarrow{f}$) pseudo-supervised model using the pseudo-parallel ($\{M_{de}, M'_{hsb}\}$) data by inducing Lample et al. (2018a) style noise (word drop, word shuffle and word blank) upon the input data.

```
--mass_steps 'de,hsb'
--encoder_only false
--emb_dim 1024 --n_layers 6
--n_heads 8 --dropout 0.1
--attention_dropout 0.1
--gelu_activation true
--tokens_per_batch 3000
--optimizer adam_inverse_sqrt,
  beta1=0.9,beta2=0.98,lr=0.0001
--word_mass 0.5 --min_len 5
```

Table 2: `MASS` pretraining parameters

4.1 MASS Pretrain and Finetune

We use the `MASS` toolkit[9] to pretrain a cross-lingual language model using the masked sequence to sequence objective. Initially, the corpus are segmented into subword units using BPE(Sennrich et al., 2016b). A joint BPE is learnt over the monolingual data of both the languages (German and Upper Sorbian) and the vocabulary is limited to 60,000 shared vocabulary tokens.

MASS Pretraining: The BPE tokenized monolingual data is used to pretrain the encoder and decoder jointly by the cross lingual MASS objective and the training is done for 100 epochs. The parameters for the `MASS` pretraining is shown in Table 2.

MASS Fine-tuning: The pretrained model is capable to generate translations but it is merely a copy task. So, in order to make the model more robust, it is further fine-tuned using the loss objective of back-translation. The fine-tuning is halted after the 10th epoch before being converged due to resource limitation. The parameters for fine-tuning is listed in Table 3.

4.2 Pseudo-Supervised NMT

We follow Marie et al. (2019) style of using the pseudo-parallel data generated from a previous

[1] `http://data.statmt.org/news-crawl/de/`

[2] `http://www.statmt.org/wmt20/unsup_and_very_low_res/`

[3] `http://www.statmt.org/wmt20/unsup_and_very_low_res/train.hsb-de.hsb.gz`

[4] The parallel side of Upper Sorbian is allowed for Unsupervised task.

[5] We use *newstest2020* test set for the submission.

[6] `http://www.statmt.org/wmt20/unsup_and_very_low_res/devtest.tar.gz`

[7] `https://github.com/moses-smt/mosesdecoder`

[8] Both Czech and Upper Sorbian belongs to Western Slavic language branch.

[9] `https://github.com/microsoft/MASS`

```
--bt_steps 'de-hsb-de,hsb-de-hsb'
--encoder_only false
--emb_dim 1024 --n_layers 6
--n_heads 8 --dropout 0.1
--attention_dropout 0.1
--gelu_activation true
--tokens_per_batch 2000
--optimizer adam_inverse_sqrt,
  beta1=0.9,beta2=0.98,lr=0.0001
--eval_bleu true
```

Table 3: MASS finetuning parameters

model to train a forward pseudo-supervised model. In our case, we first generate a synthetic data (M'_{hsb}) from the source monolingual data (M_{de}) using beam search decoding with a beam size of 10 from the MASS fine tuned model. Unlike Marie et al. (2019) where back translation was applied, we use forward translation from the source side monolingual (He et al., 2020) data to generate synthetic data. The synthetic data is detokenized, and we learn a joint subword BPE from the raw M_{de} and M'_{hsb} using Sentencepiece (Kudo and Richardson, 2018) and limit the shared vocabulary to 10 **K** units.

Noisy Pseudo-Supervised NMT: We add perturbations or noise, specifically we apply word dropout, word shuffle and word blank to our synthetic data. This kind of perturbation is found to be effective for overcoming the local minima by enforcing local smoothness (He et al., 2020; Shen et al., 2019). We train our pseudo-supervised NMT in a pseudo self-training approach by leveraging the source side monolingual data. This self-training is partial in the sense that we only use the pseudo-parallel data which lacks any sort of real labelled data for a single iteration.

The pseudo-supervised NMT is trained from scratch using Fairseq (Ott et al., 2019) toolkit[10] i.e, we do not use the previous models weights rather we apply random weight initialization for our new model. The model is trained for 300 **K** update steps. We follow Guzmán et al. (2019) style transformer architecture of 5 encoder and decoder layers, 512 embedding dimension, the feed-forward hidden dimension is 2048 with 4 multi-head attentions[11]. The rest of the parameters are listed in Table 4. We

```
--encoder-normalize-before
--decoder-normalize-before
--dropout 0.3 --relu-dropout 0.3
--attention-dropout 0.3
--label-smoothing 0.2
--criterion label_smoothed_
  cross_entropy
--weight-decay 0.0001
--lr-scheduler inverse_sqrt
--min-lr 1e-9 --max-tokens 4000
--warmup-updates 4000
--warmup-init-lr 1e-7
--optimizer adam --lr 0.0005
--adam-betas '(0.9, 0.98)'
--share-all-embeddings
```

Table 4: Pseudo-supervised NMT training parameters

make our primary submission of the test source generated using a beam search decoding with beam size of 5 and a length penalty of 1.2.

5 Result

The official automatic evaluation uses the the following metrics: BLEU (Papineni et al., 2002), TER (Snover et al., 2006), BEER (Stanojević and Sima'an, 2014), and CharactTER (Wang et al., 2016). Our primary submission (NITS-CNLP), the pseudo-supervised NMT achieves a cased BLEU of 15.4 and 15.8 as the uncased BLEU score on the *newstest2020* blind-test data. The scores are reported in Table 5. We also present the sample input-output of our primary system (NITS-CNLP) from two randomly selected test sentences from the matrix[12] in Table 6. We also report the Sacrebleu score of our various settings with the released test set (non blind test) in Table 7.

6 Conclusion

We report here the system description for our submission to the WMT 2020 shared task of Unsupervised MT for German-Upper Sorbian language pair. We submit our pipelined architecture of masked sequence to sequence pretraining along with finetuning and a pseudo-supervised model in German to Upper Sorbian direction. We observe that the performance of an unsupervised model improves significantly over the base MASS pretraining and

[10] https://github.com/pytorch/fairseq

[11] We have used 4 attention heads instead of 8 as in Guzmán et al. (2019)

[12] http://matrix.statmt.org/matrix/output/1920?run_id=7785

System	BLEU	BLEU-cased	TER	BEER 2.0	CharactTER
NITS-CNLP	15.8	15.4	0.668	0.489	0.604

Table 5: BLEU, BLEU-cased, TER, BEER 2.0 and CharactTER scores of our final primary system NITS-CNLP for the German → Upper Sorbian language using blindtest (newstest2020).

Source-1	Möchten Sie erfahren, wie sich bei uns die Unterrichtsräume mit Leben füllen?
Reference-1	Chceće wědźeć, kak so pola nas wučbne rumnosće ze žiwjenjom pjelnja?
NITS-CNLP	Časće zhonić, kak so pola nas wučbnych rumow z žiwami čuje?
Source-2	Rächt euch nicht selbst, sondern gebt Raum dem Zorn Gottes.
Reference-2	Njewjećće so sami, ale dajće městno Božemu hněwu.
NITS-CNLP	Njech wam sam, ale pomha rumnosć Božeje służby.

Table 6: Sample input-output excerpted from the matrix primary submission of NITS-CNLP.

System	BLEU
MASS-PT	2.3
MASS-FT	8.1
PSNMT	14.5

Table 7: BLEU, scores of our three systems using the released test set: MASS-pretrain (MASS-PT), MASS-finetune (MASS-FT) and Pseudo Supervised NMT (PSNMT) for German → Upper Sorbian language.

finetuning after using the synthetic data to train a pseudo-supervised model using a very naive way of self-training i.e, we have just used a single iteration of our forward training. Synthetic data is the *de-facto* for any modern semi-supervised MT system and in this experiment we show that synthetic data in an unsupervised MT is effective and also emphasised the importance of a pseudo-supervised MT model as a refinement step to an unsupervised MT.

References

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2017. Learning bilingual word embeddings with (almost) no bilingual data. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 451–462, Vancouver, Canada. Association for Computational Linguistics.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018a. Unsupervised statistical machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3632–3642, Brussels, Belgium. Association for Computational Linguistics.

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. 2018b. Unsupervised neural machine translation. In *Proceedings of the Sixth International Conference on Learning Representations*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111, Doha, Qatar. Association for Computational Linguistics.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2017. Word translation without parallel data. *arXiv preprint arXiv:1710.04087*.

Francisco Guzmán, Peng-Jen Chen, Myle Ott, Juan Pino, Guillaume Lample, Philipp Koehn, Vishrav Chaudhary, and Marc'Aurelio Ranzato. 2019. The FLORES evaluation datasets for low-resource machine translation: Nepali–English and Sinhala–English. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6098–6111, Hong Kong, China. Association for Computational Linguistics.

Junxian He, Jiatao Gu, Jiajun Shen, and Marc'Aurelio Ranzato. 2020. Revisiting self-training for neural sequence generation. In *Proceedings of ICLR*.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1700–1709, Seattle, Washington, USA. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *Advances in Neural Information Processing Systems (NeurIPS)*.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018a. Unsupervised machine translation using monolingual corpora only. In *International Conference on Learning Representations (ICLR)*.

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018b. Phrase-based & neural unsupervised machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5039–5049, Brussels, Belgium. Association for Computational Linguistics.

Benjamin Marie and Atsushi Fujita. 2018. Unsupervised neural machine translation initialized by unsupervised statistical machine translation. *arXiv preprint arXiv:1810.12703*.

Benjamin Marie, Haipeng Sun, Rui Wang, Kehai Chen, Atsushi Fujita, Masao Utiyama, and Eiichiro Sumita. 2019. NICT's unsupervised neural and statistical machine translation systems for the WMT19 news translation task. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 294–301, Florence, Italy. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Shuo Ren, Zhirui Zhang, Shujie Liu, Ming Zhou, and Shuai Ma. 2019. Unsupervised neural machine translation with smt as posterior regularization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 241–248.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Jiajun Shen, Peng-Jen Chen, Matt Le, Junxian He, Jiatao Gu, Myle Ott, Michael Auli, and Marc'Aurelio Ranzato. 2019. The source-target domain mismatch problem in machine translation. *arXiv preprint arXiv:1909.13151*.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of association for machine translation in the Americas*, volume 200. Cambridge, MA.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. Mass: Masked sequence to sequence pre-training for language generation. In *International Conference on Machine Learning*, pages 5926–5936.

Miloš Stanojević and Khalil Sima'an. 2014. BEER: BEtter evaluation as ranking. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 414–419, Baltimore, Maryland, USA. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Weiyue Wang, Jan-Thorsten Peter, Hendrik Rosendahl, and Hermann Ney. 2016. Character: Translation edit rate on character level. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 505–510.

Adobe AMPS's Submission for Very Low Resource Supervised Translation Task at WMT20

Keshaw Singh
AI/ML Platform & Solutions, Adobe Inc.
Bengaluru, India
kessingh@adobe.com

Abstract

In this paper, we describe our systems submitted to the very low resource supervised translation task at WMT20. We participate in both translation directions for Upper Sorbian-German language pair. Our primary submission is a subword-level Transformer-based neural machine translation model trained on original training bitext. We also conduct several experiments with backtranslation using limited monolingual data in our post-submission work and include our results for the same. In one such experiment, we observe jumps of up to 2.6 BLEU points over the primary system by pretraining on a synthetic, backtranslated corpus followed by fine-tuning on the original parallel training data.

1 Introduction

This paper describes our submissions to the shared task on Very Low Resource Supervised Machine Translation at WMT 2020. The task involved a single language pair: Upper Sorbian-German. We submit supervised neural machine translation (NMT) systems for both translation directions, Upper Sorbian→German and German→Upper Sorbian.

NMT models (Sutskever et al., 2014; Bahdanau et al., 2015; Cho et al., 2014a) have achieved state-of-the-art performance on benchmark datasets for multiple language pairs. A big advantage of such systems over phrase-based statistical machine translation (PBSMT) (Koehn et al., 2003) models is that they can be trained end-to-end. The bulk of the development, however, has been limited to a handful of high-resource language pairs. The primary reason is that training a well-performing NMT system requires a large amount of parallel training data, which means a lot of equivalent investment in terms of resources. Koehn and Knowles (2017) show that when compared to PBSMT approaches, NMT models need more training data to achieve the same level of performance.[1] One of the most popular ways to increase the amount of parallel training data for supervised training is backtranslation (Sennrich et al., 2016a). We utilize this approach to improve upon the performance of our baseline models.

All of our systems follow the Transformer architecture (Vaswani et al., 2017). Our primary system is a supervised NMT model trained on the original training bitext. We also report our results on experiments with backtranslation, which were completed post the shared task and hence not a part of our primary submissions. We use the backtranslated data in two distinct ways - as a standalone parallel corpus, and to create a combined parallel corpus by mixing in a 1:1 ratio with the provided training data. We also report the performance of fine-tuned models originally trained only on the backtranslated data. In the following sections, we begin by briefly describing the Transformer architecture and backtranslation. We then discuss our experimental setup as well as our experiments with backtranslation. We conclude with a discussion of our results and possible future work.

2 Related Work

The Transformer model is the dominant architecture within current NMT models due to its superior performance on several language pairs. While still a sequence-to-sequence (Sutskever et al., 2014) model composed of an encoder and a decoder, Transformer models are highly parallelizable thanks to being composed purely of feed-forward and self-attention layers rather than recurrent layers (Hochreiter and Schmidhuber, 1997; Cho et al., 2014b). The reader is encouraged to read the original paper (Vaswani et al., 2017) to gain a deeper understanding of the model. We adopt the Transformer base architecture available under the

[1] As measured by BLEU score (Papineni et al., 2002).

Proceedings of the 5th Conference on Machine Translation (WMT), pages 1144–1149

fairseq[2] (Ott et al., 2019) library for all our models.

However, NMT models are known to be data-hungry (Koehn and Knowles, 2017); their performance improves sharply with the availability of more parallel training data. Except for a few language pairs (e.g. English-German), most have little to no such data available. On the other hand, a far greater number of languages have a decent amount of monolingual data available online (e.g. Wikipedia).

To address this issue of lack of parallel data, Sennrich et al. (2016a) introduced the concept of back-translation. It involves creating a synthetic parallel corpus by translating sentences from the target-side monolingual data to the source language and making corresponding pairs. A baseline target→source model (PBSMT or NMT), trained with limited data, is generally used for this purpose. It enables the use of large corpora of monolingual data for several languages, the size of which is typically orders of magnitude larger than any corresponding bitext available. What is notable is that only the source-side data is synthetic in such a scenario and the target-side still corresponds to original monolingual data.

Some studies (Poncelas et al., 2018; Popel, 2018) have investigated the effects of varying the amount of backtranslated data as a proportion of the total training corpus, including training only on the synthetic dataset as a standalone corpus. We follow some of the related experiments conducted by Kocmi and Bojar (2019) on Gujarati-English (another low-resource pair) with a few exceptions. Besides, we also report performance when pretraining solely on the synthetic corpus following by fine-tuning on either original or mixed data. While not quite the same, one could think of this approach as having some similarities with transfer learning (Zoph et al., 2016) as well as domain adaptation (Luong and Manning, 2015; Freitag and Al-Onaizan, 2016) for machine translation. There has also been work on using sampling (Edunov et al., 2018) for generating backtranslations, but we stick to using beam search in this work.

3 Experimental Setup

3.1 Dataset

We used the complete parallel training corpus for our primary systems. In addition, we also made use of monolingual data from each language for two purposes - learning Byte Pair Encodings (BPE) (Sennrich et al., 2016b) and backtranslation. For Upper Sorbian (hsb), we used the monolingual corpora provided by the Sorbian Institute and by the Witaj Sprachzentrum. To control the quality of the backtranslated data, we chose not to use the data scraped from the web. For the German (de) side, we made use of the News Crawl[3] 2009 dataset, as it is large enough to satisfy the requirements for our experiments.

3.2 Data Preprocessing

Source	No. of sentences
hsb-de, bitext	58,389
hsb, monolingual	540,994
de, monolingual	2,000,000

Table 1: Processed training data.

Moses toolkit (Koehn et al., 2007) was used for tokenization and punctuation normalization for all data. Before doing any additional preprocessing, we learned separate truecaser models using the toolkit. For this purpose, we took first 500K sentences from each of the monolingual corpora and aggregated them with the corresponding portion from the training bitext. After tokenizing and truecasing, we joined the parallel training corpus with the same monolingual data. We learned joint BPE[4] with 32K merge operations over this corpus and applied them to the parallel training data to get vocabularies for each language. Additionally, we used the `clean-corpus-n.perl` script within Moses to filter out sentences from the parallel corpus with more than 250 subwords as well as sentence length ratio over 1.5 in either direction. Final corpus statistics are presented in Table 1.

3.3 Training

Our primary system is a Transformer base model, trained on the parallel training corpus for both translation directions till 60 epochs. We keep most of the hyperparameters to their default values in fairseq. More precisely, we chose Adam (Kingma and Ba, 2015) as the optimizer and Adam betas were set to 0.9 and 0.98, respectively. The maximum number of tokens in each batch was set to 4096. Learning rate was set to 0.0005, with an inverse squared

[2] https://github.com/pytorch/fairseq

[3] http://data.statmt.org/news-crawl/de/

[4] https://github.com/glample/fastBPE

root decay schedule and 4000 steps of warmup updates. Label smoothing was set to 0.1 and dropout to 0.3. Label-smoothed cross-entropy was used as the training criterion.

We trained all our models for a fixed number of epochs, determined separately for each system, and chose the last checkpoint for reporting BLEU (Papineni et al., 2002) scores on the test sets.

All training was done using a single NVIDIA P100 GPU. Due to the small amount of parallel training data, each epoch of training took about 90 seconds on average for the primary system.

4 Additional Backtranslation Experiments

In this section, we report our post-submission work on using monolingual data for backtranslation. We took the raw monolingual data that we describe in Section 3.1 and backtranslated with our primary submission models for the respective translation directions, i.e., hsb→de for Upper Sorbian data and de→hsb for German data. We used `fairseq-generate` function with a beam size of 5 for this purpose. Once again, we limited the number of subwords in each sentence to 250. Finally, we took all sentence pairs for backtranslated Upper Sorbian corpus and the first two million sentence pairs for the German corpus. Table 1 indicates the size of the backtranslated corpora by original language. For further experiments, we name the datasets as follows:

- *auth*: Processed original training data.
- *synth*: Backtranslated de→hsb and hsb→de corpora.
- *mixed*: Augmented training data obtained by mixing *auth* with a portion of *synth* in 1:1 ratio, providing a total of 116,778 sentence pairs.

We define the following systems for making use of the backtranslated data. Note that the first system only differs from the primary system in the number of training epochs completed.

- *auth-from-scratch*: This system has the same settings as the primary system. It was trained on the *auth* corpus till 80 epochs (as opposed to 60 for primary).
- *mixed-from-scratch*: We trained models on *mixed* data from scratch for 40 epochs.[5]
- *synth-from-scratch*: Models were trained only on the *synth* datasets. To adjust for the difference in the size of the respective backtranslated corpora, we trained hsb→de system for 10 epochs and de→hsb system for 30 epochs.
- *synth-auth-finetune*: We took the models trained via the previous system and fine-tuned them on *auth* data for 20 epochs in each translation direction.
- *synth-mixed-finetune*: Same as the last model, except that fine-tuning was done on *mixed* data.

Fine-tuning was carried out by loading pretrained checkpoints and adding extra training flags in `reset-optimizer` and `reset-lr-scheduler`.

5 Results

The systems were evaluated on the blind test set (newstest2020) using automated metrics; no human evaluation was done. Table 2 shows cased BLEU scores for various systems. Our primary systems achieved a BLEU score of 47.6 for Upper Sorbian→German and 45.2 for German→Upper Sorbian translation. We achieved an improvement of 0.3 and 0.4 BLEU points, respectively, by training further till 80 epochs in each direction. We also evaluated a third system, *synth-auth-finetune*, as described in Section 4, which provided a jump of 2.6 points in BLEU score over the primary system for Upper Sorbian→German and 2.5 for German→Upper Sorbian.

In addition to evaluating on blind test sets, we also report BLEU scores on the development test set in the same table. Two outcomes are worth highlighting:

- Model trained only on *synth* data for German→Upper Sorbian translation matched the performance of a similar model trained on the authentic bitext.
- Best results were obtained by fine-tuning a model trained on *synth* data with either *auth* or *mixed*.

[5]We trained further till 60 epochs, but observed no improvement in BLEU scores.

System	Dataset	Epochs	newstest2020	devtest
			hsb→de	
Primary*	*auth*	60	47.6	-
auth-from-scratch	*auth*	80	47.9	45.6
mixed-from-scratch	*mixed*	40	-	45.7
synth-from-scratch	*synth*	10	-	38.0
synth-auth-finetune	*+auth*	20	**50.2**	**49.6**
synth-mixed-finetune	*+mixed*	20	-	48.3
			de→hsb	
Primary	*auth*	60	45.2	-
auth-from-scratch	*auth*	80	45.6	46.4
mixed-from-scratch	*mixed*	40	-	47.4
synth-from-scratch	*synth*	30	-	46.5
synth-auth-finetune	*+auth*	20	**47.7**	49.0
synth-mixed-finetune	*+mixed*	20	-	**49.6**

Table 2: BLEU scores for the blind test set (newstest2020) and the development test set. Bold values in a column indicate the best scores among the evaluated systems. + Additional fine-tuning for models trained with backtranslated corpora. * Only the primary systems were evaluated before deadline.

The second result is notable since the regime of pretraining followed by fine-tuning improves the BLEU scores by up to 4 points on this test set when compared to training only on the original bitext. Moreover, while the model trained on *synth* was not able to match the performance of that trained on *auth* for Upper Sorbian→German, it still provides the same benefits as German→Upper Sorbian model when fine-tuned further. Looking at the small improvements achieved by using only the *mixed* corpus for training, increasing its size by combining upsampled *auth* data with more *synth* data might lead to even further jumps in the BLEU scores.

6 Conclusion

In this paper, we described our Transformer model for supervised machine translation for Upper Sorbian-German language pair. We take note of relatively high BLEU scores achieved by our primary systems (and those of other participants) on this low-resource language pair, which could relate to the high quality of the training corpus. We also report results and takeaways from several experiments with backtranslated data completed post the shared task. A key result is matching the performance of a system trained on the original bitext with one trained on a limited amount of synthetic, backtranslated data. Domain mismatch and a difference in the quality of monolingual corpus might have prevented the system from achieving a similar result in the other direction. We notice big improvements in performance over the primary systems by following a "pretraining then fine-tuning" regime.

An interesting future work would be to measure the applicability of this approach to other low-resource language pairs. Additional systems could be added as well. For instance, models trained on *mixed* data and fine-tuned on *auth* data might provide a meaningful comparison. Prior work (Ding et al., 2019) has shown that the number of BPE merge operations has a significant effect on the performance of NMT systems. This work was pointed out during the review process and should be an avenue for further improvement of the model performance.

Acknowledgments

The author would like to thank his manager for supporting this project, and the anonymous reviewers for their thoughtful comments which helped improve the presentation of this work.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015*, San Diego, California, USA.

Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014a. On the properties of neural machine translation: Encoder–decoder

approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111, Doha, Qatar. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014b. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Shuoyang Ding, Adithya Renduchintala, and Kevin Duh. 2019. A call for prudent choice of subword merge operations in neural machine translation. In *Proceedings of Machine Translation Summit XVII Volume 1: Research Track*, pages 204–213, Dublin, Ireland. European Association for Machine Translation.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500, Brussels, Belgium. Association for Computational Linguistics.

Markus Freitag and Yaser Al-Onaizan. 2016. Fast domain adaptation for neural machine translation. *arXiv preprint arXiv:1612.06897*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015*, San Diego, California, USA.

Tom Kocmi and Ondřej Bojar. 2019. CUNI submission for low-resource languages in WMT news 2019. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 234–240, Florence, Italy. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39, Vancouver. Association for Computational Linguistics.

Philipp Koehn, Franz J. Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 127–133.

Minh-Thang Luong and Christopher D Manning. 2015. Stanford neural machine translation systems for spoken language domains. In *Proceedings of the International Workshop on Spoken Language Translation*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

A Poncelas, D Shterionov, A Way, GM de Buy Wenniger, and P Passban. 2018. Investigating backtranslation in neural machine translation. *arXiv preprint arXiv:1804.06189*.

Martin Popel. 2018. Machine translation using syntactic analysis. *Univerzita Karlova*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, pages 3104–3112. Curran Associates, Inc.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008. Curran Associates, Inc.

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer learning for low-resource neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1568–1575, Austin, Texas. Association for Computational Linguistics.

On the Same Page? Comparing Inter-Annotator Agreement in Sentence and Document Level Human Machine Translation Evaluation

Sheila Castilho
School of Computing
Adapt Centre
Dublin City University
sheila.castilho@adaptcentre.ie

Abstract

Document-level evaluation of machine translation has raised interest in the community especially since responses to the claims of "human parity" (Toral et al., 2018; Läubli et al., 2018) with document-level human evaluations have been published. Yet, little is known about best practices regarding human evaluation of machine translation at the document-level. This paper presents a comparison of the differences in inter-annotator agreement between quality assessments using sentence and document-level set-ups. We report results of the agreement between professional translators for fluency and adequacy scales, error annotation, and pair-wise ranking, along with the effort needed to perform the different tasks. To best of our knowledge, this is the first study of its kind.

1 Introduction

Increasing efforts have been made in order to add discourse into neural machine translation (NMT) systems. However, the results reported for those attempts are somehow limited as the evaluation is still mostly performed at the sentence level, using single references, which are not able to recognise the improvements of those systems. The state-of-the-art automatic evaluation metrics have been shown to underestimate the quality of NMT systems (Shterionov et al., 2018), and the suitability of these metrics for document-level systems has also been criticised (Smith, 2017). For that reason, document-level human evaluation of machine translation (MT) has raised interest in the community recently as it enables the assessment of suprasentential context.

In a survey with native speakers, Castilho et al. (2020) tested the context span for the translation of 300 sentences in three different domains, namely reviews, subtitles, and literature. Over 33% of the sentences tested were found to require more context than the sentence itself to be translated, and from those, 23% required more than two previous sentences to be properly translated. Ambiguity, terminology, and gender agreement were the most common issues found to hinder translation. Moreover, differences in issues and context span were found between domains. This shows that document-level evaluation enables the assessment of textual cohesion and coherence types of errors which are impossible at times to recognise at sentence level.

Recent attempts to assess quality at the document-level were described in Toral et al. (2018) and Läubli et al. (2018) who independently reassessed the bold claims of MT 'achieving human parity' and found that the lack of extra-sentential context has a great effect on quality assessment, and pointed to a failure of the current best practices in MT evaluation. Toral et al. (2018) used consecutive single sentences to rank translations by two MT systems and a human reference. They found that the evaluators were able to better assess the translations when provided with more context, and moreover, inter-annotator agreement (IAA) between professional translators was higher than that between non-experts. However, this methodology does not discriminate sentence vs document-level set up as single sentences are shown consecutively.

Läubli et al. (2018) used pairwise rankings of fluency and adequacy to evaluate the quality of MT vs human translation (HT) for document-level texts. The methodology consists of translators choosing the 'best' translation in terms of i) adequacy and ii) fluency, that is, instead of choosing on a scale on how fluent or adequate the translations are, the raters just choose the 'best' one. Although not reporting IAA in the main paper, the authors report some IAA scores in the appendix of that work, showing that for fluency, document-level set up has

Proceedings of the 5th Conference on Machine Translation (WMT), pages 1150–1159

higher IAA than for sentence set-up, but the opposite for adequacy. However, while this evaluation methodology seems appropriate when comparing different translations, it would not be feasible when evaluating a single MT system.

After these two papers were published, for the first time the WMT19 attempted a document-level human evaluation for the *news* domain. In that year, the direct-assessment (DA) task asked crowdworkers to give a score (0-100) regarding the accuracy of the translated sentence, where only one MT output is shown each time (no comparison with other MT system). However, conventional Kappa cannot be using with DA to measure IAA, and so consistency is measured instead, where raters have to pass some quality control criteria (Barrault et al., 2019).

In light of this, a comparison of IAA between quality assessments on sentence and document-level set-ups is needed in order to determine which set-up results in most reliable evaluation. This study presents a small-scale comparison on the differences in IAA between these two methodologies. To the best of our knowledge, this is the first paper to compare IAA for sentence vs document-level set-ups using the state-of-the-art MT evaluation metrics, namely fluency and adequacy scales,[1] error mark-up and pairwise ranking, along with reporting effort indicators.

We provide a detailed description of the experimental methodology in Section 2. Following, we report results in Section 3 on the agreement between professional translators for fluency and adequacy scales, error annotation, and pair-wise ranking (in 3.1), along with the effort needed to perform the different tasks (in 3.2). We discuss our results and draw conclusions in Section 4, and point out directions for future work.

2 Methodology

Five professional English (EN) to Brazilian Portuguese (PT-BR) translators were hired to perform the evaluation in terms of (1) fluency, adequacy, and error mark-up using the PET tool (Aziz et al., 2012); and (2) pairwise ranking using Google spreadsheet. The choice of PET was due to the fact that the tool is a free toolkit, easy to use, with its UI resembling translation tools, and it is able to handle different evaluations while logging time spent on tasks.

The evaluation was carried out in two scenarios: (A) evaluation at the sentence level, showing randomised sentences, one at a time, one score per sentence, and (B) evaluation at a document level, showing randomised documents, one document at a time, one score per document. After each scenario was complete, translators answered a post-task questionnaire about the evaluation and their perceived effort.

	Test Set 1	Test Set 2
Av. Sentence Length (WPD)	316	344
Av. Sentence Length (WPS)	20	21
Av. Sentence Count (SPD)	15	15
Total Words	10135	11019
Total Sentences	500	500
Total Docs	32	32

Table 1: Statistics for the test sets used, where average sentence length is calculated as words per document - WPD - (scenario B), words per sentence - WPS - (scenario A), and average sentence count is calculated as sentence per document - SPD.

	Test Set 1		Test Set 2	
	Source	Translation	Source	Translation
Flesch	47.9	57	50	55
TTR	0. 26	0.27	0.25	0.27

Table 2: Type-token Ratio (TTR) and Flesch Reading Ease Scores for both source and translated sides of Test Sets.

Corpus - We used the English corpora from WMT `newstest2019`, which has an average document length of 17 sentences (minimum 4 sentences, maximum 30 sentences). In total, 64 full documents were selected (32 per scenario) with 1000 sentences (500 per scenario). We made sure that both scenarios are comparable in terms of sentence and document length, as well as in terms of readability and lexical variation. Results on statistics of both data sets in Table 1, along with results for Type-Token Ratio and Flesch Reading Ease Scores in Table 2 (for both source and translated versions), show that, in fact, both test sets and translations are comparable. This suggests that any variation on the assessments is unlikely to be because the two test sets are overly distinctive.

The corpus was then translated from EN into PT-BR with Google Translate (for adequacy, fluency, and error mark-up) and also with DeepL for the ranking pairwise comparison.[2]

[1] It is important to notice that Läubli et al. (2018) used pairwise ranking of fluency and adequacy instead of the standard Likert scale.

[2] The online translators were used between 20-26 February 2020.

Translators - Five professional translators took part in the evaluation.[3] Their professional experience ranges from 6 to 14 years, and three of them have had previous experience with translation evaluation. A warm-up task with 20 sentences was set up so translators could get acquainted with the tasks, guidelines and tools, as well as clarify any doubts about the task. Each translator evaluated 1000 sentences, 500 in each scenario and Test Set. Table 3 shows the distribution of the task for each translator. No time limit was stipulated for the translators to finish the task, but they were asked to keep track of the time needed to complete the tasks.

Translators	T1,T5	T2	T3	T4
Test Set 1 (1-500 sent.)	S_1	S_2	D_1	D_2
Test Set 2 (501-1000 sent.)	D_2	D_1	S_2	S_1

Table 3: Distribution of tasks where S is sentence level and D is document level scenarios, and 1 and 2 is the order of the tasks.

Post-task Questionnaire - The post-task questionnaire consisted of 10 statements for each scenario, assessed on a scale from 1 to 6, where 1 is a negative answer (strongly disagree/very difficult/very tiring) and 6 is an affirmative answer (strongly agree/very easy/not tiring at all). The statements were the following:

1. I was *always* able to understand the meaning of the source [sentence/texts]
2. I was *always* able to understand the meaning of the translated [single sentence/full texts]
3. I was *always* able to recognise all the problems with the translation of [single sentence/full texts]
4. I would have preferred to evaluate [full texts/single sentences] than [single sentence/full texts]
5. I would have preferred to evaluate pair of sentences than [single sentence/full texts]
6. I would have preferred to evaluate full paragraphs than [single sentence/full texts]
7. I was satisfied with the evaluation I provided for the [single sentence/full texts] job
8. Spotting errors in the each translated [single sentence/full texts] was (difficulty level)
9. Assessing the translation quality on a [single sentence/full texts] level was (difficulty level)
10. Assessing the translation quality on a [single sentence/full level] was (fatigue):

[3] At the time of submission of the paper, we had reported scores from 4 translators (T1, T2, T3 and T4). For the final submission we decided to add a 5th translator to be compared with T1&T2 in order to get a further understanding of issues observed with T1. Therefore, we keep the presentation of the scores between T1&T2 and, following, we present Fleiss Kappa scores with T1&T2&T5 as additional results.

3 Results

3.1 Inter-annotator agreement (IAA)

We compute IAA with Cohen's Kappa (Cohen, 1960) both weighted (W) and non-weighted (NW) as the most common statistics for IAA,

$$k = \frac{P(A) - P(E)}{1 - P(E)}$$

where $P(A)$ represents the proportion of times that the annotators agree, and $P(E)$ the proportion of times that the annotators are expected to agree by chance. While NW Kappa does not take into account the degree of agreement, W Kappa uses a predefined table of weights to measure the degree of disagreement between the two raters, the higher the disagreement the higher the weight. It is important to notice that in this case, W Kappa can only be calculated for adequacy and fluency as they are assessed using a Likert scale.

In addition to that, we also compute Inter-rater reliability (IRR) as the level of agreement between raters (percentage of matches), and Pearson correlation (r) between T1&T2 and T3&T4 (see Table 3). The comparison of the scenarios (sentence *vs* document) is calculated between the Test Sets (Test Set 1 & Test Set 2). We calculate IAA for all the tasks, namely adequacy, fluency, error and ranking. It is important to note that Fleiss Kappa is computed when analysing T1&T2&T5.

Due to the exploratory nature of this research, along with the small number of participants which is known to hinder the effectiveness of statistical analysis, we interpret the results gathered with these evaluations from a qualitative perspective.

3.1.1 Adequacy

Adequacy was assessed for each single sentence and full document (one score per document). Translators answered the question *"How much of the meaning expressed in the source appears in the translation?"* on a Likert scale from 1-4.[4] Table 4 shows the IAA scores for adequacy.

[4] 1. None of it, 2. Little of it, 3. Most of it, 4. All of it

Adequacy		SENTENCE	DOCUMENT
Test Set 1		T1&T2	T3&T4
Kappa	NW	0.13	0.01
	W	0.27	0.23
Pearson		0.5	0.64
p-value		0	0
IRR		47%	44%
Test Set 2		T3&T4	T1&T2
Kappa	NW	0.34	-0.06
	W	0.27	-0.12
Pearson		0.53	-0.37
p-value		0	0.03
IRR		63%	25%

Table 4: IAA for adequacy assessments for single sentences and full texts scenarios.

When looking at Test Set 1 (upper part of the table), we note that both W and NW Kappa show a higher score for single-sentence scenario. Interestingly, the difference between IAA scores for sentence and document for Test Set 2 (lower part) is very discrepant with IAA scores for document level reaching negative levels. This trend is supported by the negative correlation and the low IRR percentages. Since we have demonstrated that both test sets are comparable, we believe that translators T1 and T2 indeed disagreed on adequacy scores for the document scenario more than they did for the sentence scenario.

When adding T5 to the adequacy assessment, we see a decline in Kappa for both sentence-level and document-level scenarios, where k=0.04 and k=-0.12 respectively (Table 5) in contrast to k0.13 and k-0.06 (see Table 4). Conversely, we note an increase in IRR for both sentence and document-level scenarios, where IRR is 67% and 42% respectively (in contrast to 47% and 25%). These results draw near the results from T3&T4. Nevertheless, we note that IAA is higher when evaluations are performed in the sentence-level scenario.

Adequacy	SENTENCE	DOCUMENT
Test Set 1	T1&T2&T5	T3&T4
Kappa	0.04	0.01
IRR	67%	44%
Test Set 2	T3&T4	T1&T2&T5
Kappa	0.34	-0.12
IRR	63%	42%

Table 5: IAA for adequacy assessments for single sentences and full texts scenarios including T5.

3.1.2 Fluency

Fluency was also assessed for each single sentences and full documents (one score per document). Translators answered the question *"How fluent was the translation?"* on a Likert scale from 1-4.[5] Table 6 shows the IAA scores for fluency.

Fluency		SENTENCE	DOCUMENT
Test Set 1		T1&T2	T3&T4
Kappa	NW	0.09	0.41
	W	0.06	0.25
Pearson		0.1	0.73
p-value		0.02	0
IRR		53%	56%
Test Set 2		T3&T4	T1&T2
Kappa	NW	0.27	0.05
	W	0.34	-0.02
Pearson		0.42	-0.11
p-value		0	0.53
IRR		57%	47%

Table 6: IAA for fluency assessments for single sentences and full texts scenarios.

When looking at Test Set 1, we note that IAA is higher in the document-level scenario for both W and NW Kappa when compared to the single-sentence scenario. This is confirmed by the linear relation expressed by Pearson. This might suggest that fluency is easier to assess with full texts rather than with non-contextual sentences. However, the same is not true when looking at Test Set 2, where W Kappa even reaches negative scores in the document-level set-up. Once again, we see that the IAA differences are bigger for T1&T2 who assessed Test Set 1 in the sentence-level scenario and Test Set 2 in the document-level scenario (see more discussion on this in Section 4), which is again confirmed by the negative correlation.

Fluency	SENTENCE	DOCUMENT
Test Set 1	T1&T2&T5	T3&T4
Kappa	0.88	0.41
IRR	63%	56%
Test Set 2	T3&T4	T1&T2&T5
Kappa	0.27	-0.12
IRR	57%	50%

Table 7: IAA for fluency assessment for single sentences and full texts scenarios including T5.

When adding T5 to the fluency assessment, we see a large increase in IAA for sentence-level scenario where k=0.88 and IRR=63% (Table 7) in contrast to k=0.09 and IRR=53% (Table 6). However, we note that apart from a slight increase in IRR for the document-level scenario (50% compare to 47%), Kappa shows a decrease reaching a negative value k=-0.12. With these new results, both

[5] 1. No fluency, 2. Little fluency, 3. Near native, 4. Native

Kappa and IRR are higher when evaluations are performed in the sentence-level scenario. However, by looking at the **translator pair** T3&T4, we can see that these two translators still agreed more when judging the document-level scenario (k=0.41) than when judging the sentence-level scenario (k=0.27).

3.1.3 Error

Error annotation was performed after translators assessed fluency and adequacy. Translators were asked to select from a drop-down menu which errors they found in the MT output. Because we are only interested in the agreement level between translators (as opposed to finding out the quality of the MT system), we decided to use a simple taxonomy that consisted of four error categories: Mistranslation, Untranslated, Word Form, and Word order. Translators could also select "No errors" in case the sentence/document did not contain any error. Each sentence or document could be annotated with more than one error category, but unfortunately because PET does not allow for word-level tagging, each error category could be assigned only once. Therefore, a segment or document could be tagged as containing all the errors, some of the errors, as well as no errors (no issues), but if the translator found that the segment contained *two* mistranslation errors, for example, the mistranslation category would be assigned only once to that segment. Yet again, we believe this set-up is enough to measure agreement levels.

Error mark-up results were divided into *binary*, when raters agree whether there was an error (any type) or no errors in the sentence/document,[6] and *type*, when raters agree on the exact error type found in the sentence/document. Table 8 shows the results for IAA for the error mark-up task.

The error annotation task shows higher IAA and IRR in document-level scenario for Test Set 1, however, the low Pearson correlation score does not indicate a strong linear relation. For Test Set 2, we see that sentence-level scenario shows higher Kappa for error *type* and higher IRR, confirmed with a positive correlation. It is important to note that Kappa for the *binary* classification in the document-level scenario is 1 (marked as n/a) as translators agreed that (almost) all documents contained **at least** one error. However, Kappa penalises it as all the ratings fall into a single category.

[6]Intuitively, one might expect that at least one error will be found in a full document and so IAA will be high for document-level set-up in the binary category.

Error		SENTENCE	DOCUMENT
Test Set 1		T1&T2	T3&T4
Kappa	binary	0.28	n/a
	type	0.22	0.31
Pearson		0.21	0.08
p-value		0	0.49
IRR	binary	60%	100%
	type	50%	53%
Test Set 2		T3&T4	T1&T2
Kappa	binary	0.49	n/a
	type	0.38	0.20
Pearson		0.7	0.08
p-value		0	0.49
IRR	binary	76%	90%
	type	56%	33%

Table 8: IAA for error mark-up for single sentences and full text scenarios.

For this reason, we decided to also compute F-score for absolute error (disagreement) in the *binary* category (see Table 9).

ERROR	SENTENCE	DOCUMENT
Test Set 1	T1&T2	T3&T4
F-SCORE	60.4	100
Test Set 2	T3&T4	T1&T2
F-SCORE	76.6	93.75

Table 9: F-score for *binary* error mark-up evalaution.

F-scores show that indeed, *binary* classification is higher for the document-level scenario since we expect the full text to contain at least one error type. However, it is interesting to note that the document-level scenario for Test Set 2 presents only a 93.75 score and 90% (Table 8) since T1 and T2 disagreed in one document.

Error		SENTENCE	DOCUMENT
Test Set 1		T1&T2&T5	T3&T4
Kappa	binary	0.16	n/a
	type	0.02	0.31
IRR	binary	60%	100%
	type	56%	53%
Test Set 2		T3&T4	T1&T2&T5
Kappa	binary	0.49	-0.07
	type	0.38	-0.02
IRR	binary	76%	88%
	type	56%	50%

Table 10: IAA for error mark-up for single sentences and full text scenarios including T5.

By adding scores from T5 (Table 10), we note that IAA scores for Test Set 1 do not differ much, and document-level scenario shows higher Kappa and IRR as discussed previously. For Test Set 2, IAA scores decrease for Kappa, both for binary and error type categories. Interestingly, IRR scores for

Ranking	SENTENCE	DOCUMENT
Test Set 1	T1&T2	T3&T4
Kappa	0.36	0.22
Pearson	0.41	0.36
p-value	0	0.04
IRR	59%	56%
Test Set 2	T3&T4	T1&T2
Kappa	0.29	0.19
Pearson	0.41	0.42
p-value	0	0.01
IRR	53%	47%

Table 11: IAA for Pair-wise ranking evaluation.

the binary category also slightly decreases. This is a bit surprising as we were expecting translators to assign at least one error type to full texts. The results with T5 indicate that, annotating error at a document-level is difficult as translators cannot tag exactly what the problematic parts are.

3.1.4 Ranking

Pairwise ranking was performed between translation from Google translate and DeepL. The systems' outputs (single sentences in scenario A, and full documents in scenario B) were randomly mixed so translators would see different outputs. Translators were asked to rate their preferred translation, and ties were allowed. Table 11 shows the IAA for the ranking task.

In Test Set 1, the ranking evaluation shows higher IAA for sentence-level scenario when compared to the document-level. Test Set 2 shows document-level scenario with lower agreement as seen in previous trend.

When adding scores from T5, we can see in Table 12 that IRR scores do not change. A 0.1 point decrease in Kappa scores can be observed for Test Set 1 for the sentence-level scenario (k0.36 to k0.26), and a slight decrease in Kappa scores for the document-level scenario fro test Set 2 (k0.19 to k0.14).

Rank	SENTENCE	DOCUMENT
Test Set 1	T1&T2&T5	T3&T4
Kappa	0.26	0.22
IRR	59%	56%
Test Set 2	T3&T4	T1&T2&T5
Kappa	0.29	0.14
IRR	53%	47%

Table 12: IAA for ranking assessment for single sentences and full texts scenarios including T5.

Interestingly, when looking at the output of both systems, Google seem to prefer to drop gender markers more than DeepL, which might make the sentence less adequate in terms of specifying who is speaking but the sentence can still be very fluent.

1) **Source**: *Her* decision to pull out left everyone involved absolutely stunned.
DeepL: A decisão *dela* de se retirar deixou todos os envolvidos absolutamente atordoados.
Google: *Sua* decisão de sair deixou todos os envolvidos absolutamente atordoados.

2) **Source**: To recover *it* is a duty."
DeepL: Recuperá-*lo* é um dever".
Google: Recuperar (x) é um dever."

This might suggest that translators' personal preferences play a role in document-level evaluation as well. For instance, translators might prefer adequacy over fluency, as in example 1, or in the case when there is not enough context in the source to specify the gender or solve ambiguity, translators might prefer the drop of the gender marker (as in example 2).

3.2 Effort

The effort spent on assessing the two scenarios was calculated in two ways: i) time translators spend assessing the sentences and full texts, and ii) self-report of effort required to perform the tasks via a post-task questionnaire.

Time - The time spent on evaluating Adequacy, Fluency and error mark-up could be drawn directly from PET logs. Unfortunately, it was not possible to count time for the ranking task because the pairwise comparison was performed in Google Spreadsheet, and so no automatic log could be drawn. Although they were asked to keep track of their time while ranking the MT output, translators recorded this inconsistently. Therefore, we decide to use only the times logged in PET.

When performing the evaluation in PET, translators first had the chance to see the source and MT side by side in the post-editing window[7] and then to assess the MT output in another window. Therefore, PET logs two different times: one spent in the PE window, and one spent in the assessment window. Intuitively, one would believe that the translators would read the sentences/texts in the PE window and use the evaluation window only to

[7]It is worth noticing that the option of performing PE was disabled, so no time for any changes was counted.

Transl.		Reading	Assessing	Total
T1	Sent.	*09:29:33	*14:16:57	*23:46:30
	Doc	02:51:38	03:14:53	06:06:31
T2	Sent.	02:45:44	08:18:51	11:04:35
	Doc	03:25:39	02:08:26	05:34:05
T3	Sent.	05:42:25	03:07:27	08:49:52
	Doc	02:36:11	00:24:20	03:00:31
T4	Sent.	03:53:21	02:05:25	05:58:46
	Doc	02:41:15	01:13:46	03:55:01
T5	Sent.	00:35:22	05:43:11	6:18:33
	Doc	00:11:43	01:29:46	1:41:29

Table 13: Time spent on performing fluency, adequacy and error mark-up assessments in PET tool. (Note that T1 *times are compromised.)

give the scores. However, it is possible that some translators might have taken some time to read the source and MT in the assessment window.[8] Moreover, T1 reported that for the document-level scenario, they sometimes took a screenshot of the PE window "when the text was too long" and used it while evaluating the text in the assessment window (since the full text is not completely displayed in the PET assessment window).

For that reason, we decided to show both reading time (time spent on the PE window) and assessment time (time spent on the assessment window), and the total spent time. Table 13 shows the times spent for the task.

Unfortunately, T1 reported difficulties in carrying out the evaluation (due to COVID-19) and self-reported he was distracted while doing it, leaving the tool running mid-evaluation. For this reason, even when discarding obvious outlier times, there is a great discrepancy in the amount of time for T1 compared to the other translators: while translators had an average of 7-9 hours to complete the tasks, T1 took 23 hours to complete the task. Consequently, we decided to repeat T1's evaluation with T5 in order to see if patterns could be drawn from time spent on tasks.

Intuitively, one would expect translators to spend longer reading time for the document-level scenario compared to the sentence-level one, since full texts are longer. Furthermore, one would expect the assessing time to be longer for the sentence-level scenario since each sentence requires one assessment (500 assessments for 500 sentences), while in the document-level scenario, each text is assessed just once (32 assessments for 500 sentences). However, while T2 and T3 show longer reading time for document-level scenario, T1, T4 and T5 show lower reading time for that scenario.

Even though results for time do not seem to be a strong indicator of effort due to the PET's user interface limitation, it is interesting to note that while some translators do spend more time reading, some spend more time assessing. This might indicate that having the text available during the assessment of fluency/adequacy is essential for translators.

Post-Task Questionnaire - Translators answered the post-task questionnaire (see full statements in Section 2) after they finished all tasks in both scenarios. Table 14 show the average results for each statement (including T5's responses).

Statements	Sent.	Docs
1- understand source	5	5.4
2- understand translation	4.2	3.8
3- recognise problems	5.2	4.8
4- prefer (docs/single sent.) than (single sent./docs)	4	4.6
5- prefer pair of sentences than...	3.8	5
6- prefer full paragraphs than...	3.6	4.2
7- satisfied with evaluation	4.8	5
8- Spotting errors was (very easy - very difficult)	5.2	4.4
9- Assessing was (very easy - very difficult)	4.6	4.2
10- Assessing was (very tiring- not tiring)	3.2	1.8

Table 14: Post-questionnaire results (average). Scale range from 1 to 6 where 1 is strongly disagree/very difficult/very tiring and 6 is strongly agree/very easy/not tiring at all.

We observe a few interesting results for statements 1 and 2, where translators seem to be able to understand the meaning of the source better in the document-level scenario, but the meaning of the translation better in the sentence-level scenario. More interestingly, the average for statement 3 is slightly lower for document-level which might suggest that translators were less able to recognise all problems with the translation in the document-level scenario, likely due to the number of sentences.

Translators seem to prefer to judge single-sentences than full documents (statement 4), and, would rather evaluate sentence pairs (statement 5) or paragraphs (statement 6) than full documents.

Nevertheless, results for statements 8 indicate that translators found easier to spot errors in the full texts (which contradicts the results for statement 3). Previous work on evaluation of NMT systems (when compared to SMT) found translators find

[8] PET displays the MT and Source at the top of the assessment window.

NMT errors more difficult to identify due to its high fluency (Castilho et al., 2017) (at a sentence-level). This could be a good indication that, due to good levels of fluency in NMT systems, indeed the exhibition of full texts is more helpful for the assessment in general.

Finally, translators found the document-level scenario to be slightly easier to assess (statement 9) but much more tiring than assessing single sentences (statement 10).

4 Discussion

This paper attempts to shed light on the differences in IAA between sentence and document-level evaluation scenarios. The experiments performed with five professional translators have tested the state-of-the-art metrics commonly used to assess MT quality with humans, namely the assessment of fluency, adequacy, error mark-up and pairwise ranking (Castilho et al., 2018).

We note that when evaluating *adequacy* (Table 4) the scenario where single sentences are assessed show higher IAA for both test sets, and moreover, IAA for Test Set 2 presents the lowest IAA for the document-level scenario for all the metrics. Regarding *fluency* assessment (Table 6), document-level scenario for Test Set 1 has higher IAA but Test Set 2 the opposite is seen for Test Set 2.

In addition to scores per test sets, it is interesting also to look at the IAA scores by **translator pairs**. We observe that there is a large difference between T1&T2 who evaluated Test Set 1 in the sentence-level scenario, and Test Set 2 in document-level scenario, against T3&T4 who evaluated the opposite. T1 and T2 tend to disagree more in both Test Sets for both fluency and adequacy assessments, while T3&T4 have closer IAA scores and higher Pearson correlation. The addition of T5's assessments, reported in terms of Fleiss kappa, indicate that for the majority of the case, IAA is indeed higher when evaluation is performed at a sentence level.

Figure 1 shows examples of disagreement between translators. In example (1), T1 assessed the text as containing "little" of the meaning of the source, T2 considered it to contain "all of it", and T5 assessed it as containing "most of it" (*adequacy*). T1 comments that "many mistranslations of golf/sport terms impaired meaning" and "some unstranslated terms found ('team USA', 'singles')", while T2 thinks that the text contains "minor issues, but the meaning isn't lost", and T5 says "the meaning is compromised by the word-by-word translation". While both T1&T2 agree that the text is "near native" regarding *fluency*, T5 assess it as having "little fluency", mentioning that fluency is also "compromised by the literal translation of some terms".

For T3&T4, the disagreement in the document-level (example (2)) is not as strong. While T3 assesses it as containing "most of the meaning", T4 thinks that it contains "little of it" because "there are a couple of plays on words in the source text, a big part of the translation is lost". However both agreed that regarding fluency, example (2) has "little fluency".

We speculate that the disagreements at the document-level scenario, especially for the adequacy evaluation, might be connected to the fact that because the texts are made up of "very good", "reasonably" and "poorly" translated sentences which, together, make the text understandable to a certain level, it is harder for translators to be consistent when assigning one single score for a full text. We estimate the percentages of *adequacy* scores for the document-level scenario as follows: T1&T2&T5 show 0% for score 1 (none of it), 7.29% for score 2, 61.46% for score 3, and 31.25% for score 4 (most of it); while T3&T4 show 4.69% for score 1, 17.19% for score 2, 64.06% for score 3, and 14.06% for score 4. These results show that a great number of scores fall into the middle category which makes it difficult for a consistent evaluation on a document-level scenario. Consequently, this type of problem will be persistent when evaluating at a document-level MT systems that operates at the sentence level, because a document translated with sentence-level NMT is still a sequence of translated sentences rather than an entire document translation.

We observe that disagreements in the sentence-level are more often related to ambiguity and lack of context. In example (3), while T1 commented that the translation "failed to use football terminology" and assessed it as containing "none of the meaning", T4 and T5 assessed it as containing "all of the meaning". We speculate that T4 and T5 were unaware that the sentence was about football due to the lack of context, and did not penalise mistranslations such as 'fired' which is better translated as 'chutar' (to kick) and 'box' which should be translated as 'pequena área'. T5 even mentioned that

(1)	Ryder Cup 2018: Team USA show stomach for fight to keep hopes alive heading into Sunday singles. After three one-sided sessions, Saturday afternoon's foursomes might just have been what this Ryder Cup needed. The swinging pendulum of momentum is a completely invented sporting concept but one that players truly believe in, and never more so than at competitions like these. So where would they say the momentum is now? […]	Ryder Cup 2018: **Team USA mostra estômago para luta para manter as esperanças vivas nos singles de domingo**. Após três sessões unilaterais, o quarteto de sábado à tarde pode ter sido o que **esta** Ryder Cup precisava. **O pêndulo oscilante do momento** é um conceito esportivo completamente inventado, mas no qual os jogadores realmente acreditam, **e nunca mais do que** em competições como essas. Então, onde eles diriam que o momento é agora? [...]
(2)	Welsh AMs worried about 'looking like muppets'. There is consternation among some AMs at a suggestion their title should change to MWPs (Member of the Welsh Parliament). It has arisen because of plans to change the name of the assembly to the Welsh Parliament. AMs across the political spectrum are worried it could invite ridicule. One Labour AM said his group was concerned "it rhymes with Twp and Pwp." For readers outside of Wales: In Welsh twp means daft and pwp means poo […]	**AMs galeses** preocupados com **'parecendo muppets'**. Há consternação entre algum**as AMs** por sugestão de que seu título deve mudar para **MWPs** (membro do Parlamento de Gales). **Surgiu** por causa dos planos de mudar o nome da assembléia para o Parlamento galês. **As AMs** de todo o espectro político estão preocupad**as** com o fato de poder convidar ao ridículo. Um dos trabalhadores da AM disse que seu grupo estava preocupado "rima com Twp e Pwp". Para leitores fora do país de Gales: em galês twp significa **daft** e pwp significa cocô [...]
(3)	He then fired a beautiful through ball, leading Hazard into the box.	Ele então disparou uma bela bola cruzada, levando Hazard para dentro da caixa
(4)	It would see employees enjoy a three-day weekend - but still take home the same pay.	Veria que os funcionários desfrutariam de um fim de semana de três dias - mas ainda levariam para casa o mesmo salário.

Figure 1: Examples of disagreement between translators.

they had problems with the word "Hazard" because eve thought "it seems to be a noun as it starts with a capital letter, I could not assess whether "Hazard" is a proper noun or just a noun, due to lack of context".

Example (4) is another example of lack of context, since the pronoun "It" is impossible to identify in the sentence. While T3 decides to rely only on the context given and assess *adequacy* as "all of it" and *fluency* as "native", T4 assesses the sentence as containing "little of the meaning" and "little fluency". According to T4, "the context was not enough to translate the pronoun 'it'". This is consistent with findings in Castilho et al. (2020) where authors found that over 33% of the surveyed sentences required more context than the sentence itself to be translated. Indeed, with the context of previous sentences it is possible to identify that "it" relates to "a radical plan" and therefore the addition of "O plano veria" (the plan would see) in the translation would make it more adequate:

(+2) Jeremy Corbyn's Labour Party is to consider **a radical plan** which will see Britons working a four day week - but getting paid for five.

(+1) The party reportedly wants company bosses to pass on savings made through the artificial intelligence (AI) revolution to workers by giving them an extra day off.

(S) ***It*** *would see employees enjoy a three-day weekend - but still take home the same pay.*

Interestingly, T1, T2 and T5 who assessed this sentence in context in the document-level scenario agreed that the text was "near native" and contained "most" and "all" of the meaning. This might be another indication that fluency is better assessed at a document level.

Regarding error mark-up assessment (Table 8), even though for Test Set 1 the document-level scenario has higher IAA than the sentence-level scenario, we note that when looking at **translator pairs**, the document-level scenarios has lower IAA scores for both test sets. Looking at T3&T4 both translators agree more on the error *types* found in the sentence-level than on the document-level scenario.

Finally regarding effort, unfortunately the logged time in PET tool was not decisive, even witht he addition of T5's assessments (due to discrepant results by T1 (Table 13)). Nevertheless, we believe that the results reported here show how difficult it is to run human evaluations, especially unsupervised ones. Additionally, the lack of proper tool able to handle different MT evaluation methodologies makes the assessment even more complex. We consider that time log gathered in PET can still be useful to draw specifications to develop a MT evaluation tool able to handle different methodologies. With respect to translators' self-assessment of their effort, the results from the post-task questionnaire showed that while translators prefer to see full texts than single sentences, they would rather see sentence pairs and paragraphs than having to assess full documents. This is not surprising since evaluating at a sentence-level is is what translators are used to already. Furthermore, they find assessing a full document more tiring than the alternative.

5 Conclusions and Future Work

The present work attempts to shed light at the differences in IAA when evaluating MT at the sentence and document levels with a small scale comparison. The main key findings of this comparison is that, a document-level evaluation methodology where translators assign one score per text leads to lower levels of IAA for adequacy, ranking, and error mark-up (when compared to methodologies where translators assign one score per sentence), but it might be useful for fluency assessments. This

is consistent with (Läubli et al., 2018) findings on IAA for pairwise comparison, and previous work on NMT evaluation, where fluency proved to be harder to assess (than adequacy) in sentence-level scenarios.

Nevertheless, we also speculate that as Google Translate seems to operate on a sentence-level, a document-level evaluation of adequacy is penalised since a document can be constituted of sentences with different levels of quality. Moreover, we consider whether multiple scores per document (sentence pairs, paragraphs, and word-level error tagging) will yield higher levels of IAA when compared to the randomised sentence-level set-up for both sentence and document-levels MT systems.

Human-evaluation of MT in document-level set-ups is in its infancy, and therefore, it is essential to test which methodologies will be best suited for different tasks and domains. Future work will use more translators and different methodologies, as expressed in the post-task questionnaire and discussed above, with more specific guidelines for context-span issues found in previous works, and the development of test-sets, as well as using document-level MT systems' outputs.

Acknowledgments

I would like to thank the translators who participated in the experiment. Also a big thank you to Maja Popović and Joss Moorkens for the brainstorming sessions. This project was funded by the European Association for Machine Translation through its 2020 sponsorship of activities programme. The ADAPT Centre for Digital Content Technology (www.adaptcentre.ie) at Dublin City University is funded by the Science Foundation Ireland Research Centres Programme (Grant 13/RC/2106) and is co-funded by the European Regional Development Fund.

References

Wilker Aziz, Sheila Castilho, and Lucia Specia. 2012. PET: a Tool for Post-editing and Assessing Machine Translation. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, pages 3982–3987, Istanbul, Turkey.

Loïc Barrault, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. 2019. Findings of the 2019 Conference on Machine Translation (WMT19). In *Proceedings of the Fourth Conference on Machine Translation (WMT 19)*, pages 1–61, Florence, Italy.

Sheila Castilho, Stephen Doherty, Federico Gaspari, and Joss Moorkens. 2018. Approaches to Human and Machine Translation Quality Assessment. In *Translation Quality Assessment: From Principles to Practice*, volume 1 of *Machine Translation: Technologies and Applications*, pages 9–38. Springer International Publishing.

Sheila Castilho, Joss Moorkens, Federico Gaspari, Iacer Calixto, John Tinsley, and Andy Way. 2017. Is Neural Machine Translation the New State of the Art? *The Prague Bulletin of Mathematical Linguistics*, 108(1):109–120.

Sheila Castilho, Maja Popović, and Andy Way. 2020. On Context Span Needed for Machine Translation Evaluation. In *Proceedings of the Twelfth International Conference on Language Resources and Evaluation (LREC'20)*, Marseille, France.

Jacob Cohen. 1960. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, 20(1):37–46.

Samuel Läubli, Rico Sennrich, and Martin Volk. 2018. Has Machine Translation Achieved Human Parity? A Case for Document-level Evaluation. In *Proceedings of EMNLP*, pages 4791–4796, Brussels, Belgium.

Dimitar Shterionov, Riccardo Superbo, Pat Nagle, Laura Casanellas, Tony O'dowd, and Andy Way. 2018. Human versus Automatic Quality Evaluation of NMT and PBSMT. *Machine Translation*, 32(3):217–235.

Karin Sim Smith. 2017. On Integrating Discourse in Machine Translation. In *Proceedings of the Third Workshop on Discourse in Machine Translation*, pages 110–121.

Antonio Toral, Sheila Castilho, Ke Hu, and Andy Way. 2018. Attaining the Unattainable? Reassessing Claims of Human Parity in Neural Machine Translation. In *Proceedings of WMT*, pages 113–123, Brussels, Belgium.

How Should Markup Tags Be Translated?

Greg Hanneman and **Georgiana Dinu**
Amazon AI
{ghannema, gddinu}@amazon.com

Abstract

The ability of machine translation (MT) models to correctly place markup is crucial to generating high-quality translations of formatted input. This paper compares two commonly used methods of representing markup tags and tests the ability of MT models to learn tag placement via training data augmentation. We study the interactions of tag representation, data augmentation size, tag complexity, and language pair to show the drawbacks and benefits of each method. We construct and release new test sets containing tagged data for three language pairs of varying difficulty.

1 Introduction

The quality of machine translation (MT) has drastically improved in recent years, making MT technology more widely used than ever before in applications ranging from financial services (Nunziatini, 2019) to fashion, social media, and other user-generated content (Kosmaczewska and Train, 2019; Birch et al., 2019; Michel and Neubig, 2018), among other tasks.

A large amount of content requiring translation is not isolated plain text. Rather, it originates in the context of structured documents, using document format specifications such as HTML, Microsoft Word, PDF, etc.

Currently, the translation industry addresses the translation of structured documents by dividing the task between a translation management system (TMS) and an underlying MT system (e.g. Federico et al. (2014)). Figure 1 shows a schematic of the process. The TMS parses, manipulates, and validates the higher-level document structure. It is responsible for finding the translatable portions of the input document, performing sentence segmentation on the content, sending it to an underlying MT system for translation, and placing the result back into the document structure. For example, the TMS may pass over material contained in HTML `<script>...</script>` tags, while sending to MT the contents of `<p>...</p>` tags and the string values of `<img alt="..."/>` attributes.

Properly transferring formatting tags *within* the translatable content (bold, italic, hyperlink, superscript, etc.) remains the responsibility of the MT process rather than the TMS. The correct preservation and transfer of inline markup from source to target thus forms a crucial component of the overall quality of the MT system. An accurate placement within the segment of inline markup provides a more readable and usable document in its raw MT form, in addition to saving human time and effort if post-editing is used.

Despite the key role of the MT system in processing structured documents — both in terms of TMS expectations and in lowering translation costs — the setup of translation in the context of structure is rarely addressed in the standard MT evaluation benchmarks. They typically focus on the pure-content, string-based tasks (Joanis et al., 2013; Müller, 2017).

In this paper, we study the question of how inline markup should be represented in and processed by the MT system in order to result in the highest placement accuracy. Given the deep semantic representations and generalization abilities provided by modern neural MT systems, we design a series of experiments to test their capabilities specifically on the problem of markup transfer within the translation process. The paper makes the following contributions:

(1) We propose a technique to augment any parallel corpus with inline tags, addressing the scarcity of high-quality parallel data containing markup tags (Section 3). We show that the method results in highly accurate tag placement and can improve the accuracy of tag placement of MT models when used to augment the training data.

Proceedings of the 5th Conference on Machine Translation (WMT), pages 1160–1173
Online, November 19–20, 2020.

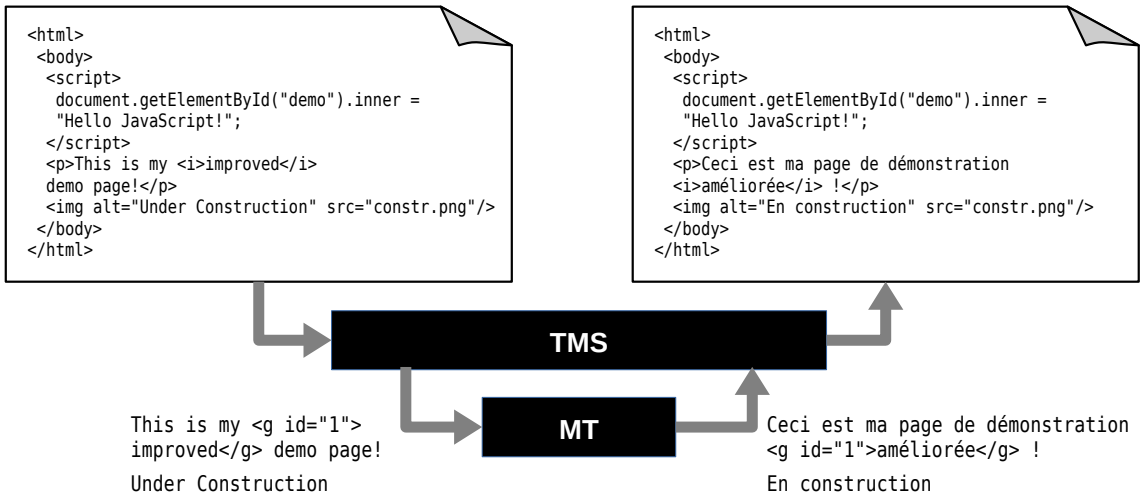


Figure 1: Translation of a structured document. A translation management system (TMS) is responsible for higher-level document structure, while the machine translation (MT) system transfers inline markup.

(2) We provide a comprehensive evaluation of several tag-handling methods (Section 4). We test the ability of neural networks to jointly learn to translate content and transfer tags from automatically generated tagged data. Within this approach, we experiment with two ways of representing tags and compare these with a baseline detag-and-project method, on language pairs of varying data sizes and difficulty (English–French, English–German, and English–Hungarian).

(3) Finally, testing of tag placement accuracy poses difficulties in terms of both data-set availability and quality metrics. For this reason, we assemble new test sets of natively tagged structured documents for the three targeted language pairs. We enhance these test sets by adding synthetically generated tagged data according to human-quality glossary entries (Section 5), and we release these test sets in order to facilitate further research on the topic. Evaluation is carried out according to standard automatic metrics, automatic detection of obvious tagging errors, and human assessments of tag placement accuracy (Section 6).

2 Related Work

Various works have addressed the inline markup problem in the context of statistical MT systems. Joanis et al. (2013) extensively summarize many such methods. Their "two-stream" approach is quite similar to our detag-and-project baseline, relying on phrase and word alignments and carefully designed tag transfer rules to re-insert inline markup into the translated content. Though the accuracy of this approach is tested in a small human evaluation, it is explicitly not compared to any other method of tag representation.

Müller (2017) addresses the same question, comparing five different varieties of detag-and-project and mask-based approaches. The work concludes that the detag-and-project approach of Joanis et al. (2013) performs the best in complicated tagging scenarios, while masking can be a strong approach in simplistic cases.

Moving to neural MT, Hashimoto et al. (2019) experiment with "raw" tags that are left unmodified in the input. The authors test a constrained beam search that restricts the decoder to outputting all and only the tags actually present in the source while maintaining XML well-formedness. They also introduce a pointer mechanism (See et al., 2017) to promote copy-through of tags and other non-translatable content. These restrictions are shown to improve output, but no other tag representation aside from raw tags is tested.

We thus position our work as spanning and unifying elements of these previous studies, directly comparing the major tag representation methods with a minimum of other modeling changes in a state-of-the-art neural MT setting. Additionally we introduce a data augmentation technique that removes the dependence on pre-existing tagged training data assumed by the approaches proposed by e.g. Hashimoto et al. (2019).

In this context, we also note complementary work on the implicit learning of structure by sequence-to-sequence models. Target-side struc-

ture expressed as content, without special annotations, has been used in string-to-tree syntax-based MT by Nădejde et al. (2017) and Aharoni and Goldberg (2017); the latter group notes that well-formed parse trees are generated 99.5% of the time. The masking method of passing abstracted information between input and output has been applied in MT to non-translatables such as numbers, URLs, named entities, etc. (Crego et al., 2016; Post et al., 2019). It has been shown to improve automatic metric scores, with the masks correctly appearing in the MT output a very high percentage of the time (Murakami et al., 2019; Bérard et al., 2019). Source factors, or additional input streams provided in lockstep with the content to translate (Sennrich and Haddow, 2016), are one mechanism for signaling structural information in the input while leaving it accessible to the model. It has been used to augment an MT system with terminology constraints (Dinu et al., 2019), and the same mechanism could be in principle applied to tags as well.

Training data augmentation is a popular way to increase model learning and robustness for specific phenomena. Of particular interest is the finding by Karpukhin et al. (2019) showing that a "balanced diet" of training data synthetically augmented with spelling errors improved model performance on natural errors, without harming performance on clean text. Our notion regarding synthetically injected and naturally occurring tags is similar.

3 Data Augmentation via Tag Injection

The vast majority of data available for MT research appears as plain text only.[1] While tagged training corpora, when available, will inevitably be limited in size, domains covered, and language availability, we propose as an alternative a general technique for injecting markup tags of controlled complexity into any parallel corpus with high accuracy.

Although not always adhering to the strict syntax of a well-defined markup language, most inline tags follow the XML standard and create a hierarchical structure within a segment by introducing either paired tags (opening and closing) or self-closing unary tags. This structure is expected to be preserved by the translation, with paired tags surrounding corresponding text fragments. In Figure 1, the italicized fragment *improved* transfers to italicized *améliorée*. We propose a method to identify such corresponding fragments in the source and target sides of parallel data and to automatically inject tags based on them.

3.1 Tag Injection

We identify corresponding fragments using the hypothesis that, if the out-of-context translation of a sentence fragment is found in the target sentence, then those text fragments are aligned. More formally, assume source segment s and target segment t are each decomposed in three substrings $(s,\ t) = (a\ b\ c,\ x\ y\ z)$, where b and y are not empty. We hypothesize that if an MT model translates b into y in isolation, then b and y are corresponding fragments and we can inject the following tag structure:

$$a\ \texttt{<t>}b\texttt{</t>}\ c$$
$$x\ \texttt{<t>}y\texttt{</t>}\ z$$

In our implementation, the search over candidate n-grams b proceeds in random order, subject to a parameter controlling the maximum span. Our preliminary experiments showed that natively tagged text is not always well-formed. For this reason we introduce a "pair damage fraction" parameter, which sets the probability that one half of the injected tag pair will be skipped. This produces the patterns a <t>$b\ c$ or $a\ b$</t> c, with analogous results on the target side. We also incorporate a fixed probability that the tag will be injected as self-closing rather than a pair, i.e. as a <t/>$b\ c$ or $a\ b$<t/> c. Finally, we allow for the injection of multiple tags into the same parallel segment via a parameter that specifies the maximum number of tag pairs to insert per segment.

This method can be adapted to any markup schema choice. For the experiments described in this paper, we choose XLIFF, a translation industry standard that uses a reduced vocabulary of tag and attribute names as an abstraction over an original document's markup language. We inject four tags in accordance with the XLIFF 1.2 specification:[2] <g>...</g> for a paired tag, <x/> for a self-closing tag, <bx/> for a tag that opens without closing, and <ex/> for a tag that closes without opening. In all cases we include a numerical id attribute, with the value starting at 1 for the leftmost tag injected in each source segment and incrementing by one for each successive tag in left-to-right

[1] A recent exception is the 17-language XML-tagged parallel corpus described by Hashimoto et al. (2019), released publicly in July 2020 concurrently with our work.

[2] http://docs.oasis-open.org/xliff/v1.2/cs02/xliff-core.html#Specs_Elem_Inline

Language	Tags	Cor	Inc	Imp	Unc
EN–DE	532	484	22	8	18
		501	17	7	7
		501	17	4	10
EN–FR	560	504	29	7	20
		504	31	5	20
		508	38	3	11
EN–HU	540	438	47	49	6
		497	32	6	5
		463	20	47	10

Table 1: Human evaluation of tag injection accuracy, with the count of tags judged as **Cor**rect, **Inc**orrect, **Imp**ossible, or **Unc**lear by each of three annotators.

source order.

Tag injection may fail in specific instances for several reasons, such as inability to find a (b, y) pair, or matched phrases overlapping with each other and blocking intended injections. We do not explicitly remove or attempt to repair any of these cases, using them instead as additional sources of variety and noise in the final tag-injected corpus.

3.2 Injection Evaluation

We test our hypothesis that the proposed procedure leads to accurate data by performing a human evaluation on the accuracy of tag placement. This experiment is carried out on a random selection of 200 sentence pairs from our training data in each of three language pairs used in our subsequent experiments: English–German (EN–DE), English–French (EN–FR), and English–Hungarian (EN–HU). (See Section 5.1 for details on the training data itself.)

We ask bilingual speakers in the relevant languages to judge whether each individual tag is correctly placed in the output, incorrectly placed (and a proper location for it can be identified), impossible to place (because no correct location exists given the target content), or too unclear to evaluate (e.g. because a placement decision depends on the semantics of the tag). Each set of sentences is independently evaluated by three different people who are not aware of how the tagged data was created.

The results of the evaluation (Table 1) show that the tag injection is accurate: using all the judgements combined, tags are correctly placed in 93.1% of cases in EN–DE, 90.2% in EN–FR, and 86.3% in EN–HU. The rates of actually wrongly placed tags are 3.5%, 5.8%, and 6.1%, respectively, with the remaining tags being judged as impossible to place or as unclear.

Pairwise inter-annotator agreement varies: judges clearly disagree (correct vs. incorrect) no more than 2.3% of the time in EN–DE and EN–FR, though up to 5.9% in EN–HU. Because of the large class imbalance, we also examine Fleiss's κ metric with all three annotators. The overall κ scores are 0.69 for EN–DE and 0.72 for EN–FR, but only 0.39 for EN–HU. In Hungarian, a larger number of tags are judged as correctly placed by one annotator but impossible to place by another, underscoring the difficulty of markup transfer between morphologically and grammatically distant languages.

4 Tag Representations

4.1 Baseline: Detag/Project

Our baseline setup does not model inline tags in any way. Instead, all markup is removed from the run-time input and reinserted into the MT output by a post-processing step. Reinsertion is carried out via *tag projection*: it uses the position of a tag in the input, a subword-level alignment model between source and target, and a set of projection heuristics to determine the analogous position of the tag in the MT output.

The necessary alignment model is built from the MT system's own training data. We use BPE (Sennrich et al., 2016) for subword creation and FastAlign (Dyer et al., 2013) for training the alignment model. At run time, we force-align the detagged BPE-level MT input to the BPE-level MT output, then convert the source-side subword indexes back to their token-level equivalents.[3] This provides a mapping between input tokens and output BPE pieces. A tag or tag pair is transferred from input to output according to this map and several hard-coded rules. Let s_i and t_i represent the ith source token or target BPE piece, respectively, in a segment of length I and J, and let $A(s_i) = \{j\}$ be the set of target indexes j of alignments induced for s_i. Then the most important projection rules are:

- A tag pair <x>...</x> spanning $s_a ... s_b$ encompasses alignments $\ell = \bigcup_{i=a}^{b} A(s_i)$ and is projected to span $t_{\min(\ell)} ... t_{\max(\ell)}$.
- A self-closing tag <x/> appearing before s_a follows alignments $\ell = A(s_a)$ and is projected to appear before $t_{\min(\ell)}$.

[3] An alternative is building and applying the alignment model on tokens instead of BPE pieces. As a practical concern, we prefer the BPE level for ease in handling non-whitespace languages and for its substantially smaller model size.

Original <bx id="1"/>As regards <g id="2">exports</g> of <g id="3">radioactive waste</g> from the Community to third countries, six Member States issued a total of 13 authorisations, representing 35 shipments.

Masked __XLF_BX,1__ As regards __XLF_OPENG,1__ exports __XLF_CLOSEG,1__ of __XLF_OPENG,2__ radioactive waste __XLF_CLOSEG,2__ from the Community to third countries , six Member States issued a total of 13 authoris@@ ations , representing 35 shipments .

Raw <bx id="1"/> As regards <g id="2"> exports </g> of <g id="@@ 3"> radioactive waste </g> from the Community to third countries , six Member States issued a total of 13 authoris@@ ations , representing 35 shipments .

Figure 2: Example of inline markup tag representation when it is included in the MT input, either by masking or in raw form.

- A tag pair with zero span or a self-closing tag appearing before s_0 is projected to appear before t_0. The same appearing after s_I is projected to appear after t_J.

Additional heuristics specify behavior for cases when the relevant alignment set is empty, when the nesting of the input tags is malformed, when an unpaired tag <x> or </x> appears without its other half, etc.

4.2 Masked and Raw Representations

We compare the above approach to two alternatives where some representation of inline markup is provided to the MT system. Tags are either masked to generic placeholder tokens or else left raw in the input. Figure 2 gives an example of each.

Aside from reducing vocabulary, masking protects the original content from subword splits and mutilations during the MT process. In our implementation, if a mask is present in the source but fails to appear in the MT output, we forcibly add it back at the end of the output; spuriously generated masks are likewise removed. We use a different placeholder name for each of the five XLIFF tag names present in our data; this treats <g> and </g> independently. We also include a sequence number in the mask token, so that the original content can be matched with the correct placeholder in the target side. Similar to the XLIFF `id` parameter, these sequence numbers start with 1 in each sentence pair and increase left to right in the source sentence. Unlike in XLIFF, our placeholder sequence numbers are incremented individually for each of the five placeholder names.

Our other alternative approach leaves the XLIFF tags raw in the input, trusting the MT system to learn their correct formatting as well as placement.

Both the masking and the raw approach rely on tags appearing often enough in the training data — not only for the MT system to learn their transfer and placement, but also to be recognized as tokens as part of the system's subword vocabulary. Since the masks are single tokens, adding self-translating examples as additional parallel data and exempting the mask tokens from BPE application is sufficient. For the raw approach, we include the same number of self-translation examples to boost the frequency count of XLIFF tag tokens above the minimum BPE frequency cutoff. As Figure 2 illustrates, however, this does not necessarily ensure that all possible tag tokens appear often enough in the training data to be recombined by BPE into full tokens: lower-numbered tags still occur more frequently than higher-numbered examples.

5 Experimental Setup

5.1 Training Data

Our training data is sourced from the Conference on Machine Translation (WMT) series of shared tasks. Since our focus is on inline tag handling rather than corpus filtering or new state-of-the-art translation quality, in some cases we have ignored especially large or noisy data sets.

For EN–DE, we begin with the training data released by the WMT 2020 news task, ignoring the Common Crawl and Paracrawl corpora and heavily filtering WikiMatrix. Our EN–FR training data comes from the 2014 news translation task; we use only Europarl, News Commentary, and UN Docs. For EN–HU, we use the single available training corpus from the 2009 translation task.[4] Our final training data comprises 5.7 million lines for EN–DE, 14.5 million lines for EN–FR, and 1.5 million

[4]Data available from http://www.statmt.org/wmt20/translation-task.html (EN–DE), http://www.statmt.org/wmt14/translation-task.html (EN–FR), and http://www.statmt.org/wmt09/translation-task.html (EN–HU).

lines for EN–HU. See Appendix A for the exact enumeration of component corpora, line counts, and a description of our filtering process.

On top of this baseline data, we experiment with various amounts of tag-injected data augmentation. We sample 1%, 2%, 5%, 10%, or 15% of the baseline training data and inject XLIFF tags into it as described in Section 3. We attempt to inject up to two pairs of tags per segment, with a max span of six tokens, pair damage fraction of 0.10, and self-closing fraction of 0.27. Candidate n-grams for tag injection are identified by looking for equivalent translations with the publicly available version of Amazon Translate as of April 2020. Lines that completely fail tag injection are discarded, not counted as part of the goal percentage, and replaced with successful lines. Each successive augmentation percentage is a strict superset of the ones before it: e.g. all the data present in the 2% corpus is also present in the 5%, 10%, and 15% settings.

5.2 Tagged Dev and Test Sets

Although we believe the tag injection technique achieves sufficiently high accuracy to be used in training data, we prefer our development and test sets to represent — as much as possible — naturally occurring tags in the source and human-quality placements for them in the target. This section describes the test sets created. They are also publicly available at https://github.com/amazon-research/mt-markup-tags.

EUR-Lex EUR-Lex[5] is the European Union's online repository of legal documents, which are provided synchronously in several structured formats and in the union's 24 official languages. We select a cohesive block of documents in Microsoft Word format, from CELEX numbers 52019DC0601 through 52019DC0680, to serve as the base of our dev and test sets. Each document is available as monolingual downloads in English, German, French, or Hungarian; several processing steps are needed to create a sentence-aligned tagged parallel corpus.

For a set of four monolingual documents, we first extract each one from Word to XLIFF format using the open-source Okapi Tikal document filter.[6] Aside from performing automatic paragraph and sentence segmentation according to pre-defined rules, the Okapi filter also converts the inline Microsoft markup to XLIFF 1.2 tags. We then check the extracted XLIFF documents for parallelism at several levels. Any document set that does not contain the same number of paragraphs across all four languages is entirely rejected. Any paragraph that does not have the same number of sentences across languages is skipped; any sentence that does not have the same set of XLIFF tags across all languages is likewise skipped. The surviving sentences form a four-way parallel corpus with inline markup tags. Each successfully extracted document is then assigned to either the dev or the test set. Sets assembled in this way are finally deduplicated to unique sentence pairs. The EUR-Lex dev set contains 1888 lines; the test set 1450.

We find, surprisingly, that a significant number of otherwise parallel segments do not contain the same inline tags across all languages. One side effect of enforcing this restriction is that the tags that are indeed parallel are biased towards trivial cases, such as an opening tag at the beginning of the sentence and a closing tag at the end. Only 11% of the sentences extracted above contain line-medial tags, and only 4% contain more than two tags per line. We mitigate this problem via the construction of two additional sets.

EUR-Lex mono We return to the unfiltered monolingual English documents assigned to the EUR-Lex test set. Without parallelism restrictions, this collection forms a much more diverse test set: after deduplication, 26% of lines contain medial tags and 13% hold more than two tags. While we are unable to use this for MT evaluation metrics that require a reference, we employ this 2525-line test set for other types of automatic and human evaluation.

Glossary We use additional EUR-Lex documents (CELEX numbers 52019DC0520 to 52019DC0599) to construct dev and test sets with synthetically introduced tags. In contrast to tag injection, however, the markup is inserted using human-curated translation glossaries. We extract and filter each set of monolingual Word documents as before, with the additional step of removing all the inline tags to obtain plain text. Given four-way parallel segments, we then search within each line for a synchronous occurrence of entries from our glossaries for EN–DE, EN–FR, and EN–HU. If

[5] https://eur-lex.europa.eu/homepage.html

[6] https://okapiframework.org/wiki/index.php/Tikal

found, the terms are surrounded by a pair of identical XLIFF <g>...</g> tags in each language. The synchronous glossary restriction reduces heavily the amount of successfully extracted and tagged sentences. On the other hand, it provides in practice 100% examples with line-medial tags, as well as an improved 22% of lines containing more than two tags. The final sizes are 286 lines for dev and 289 for test.

The complete dev set for our MT systems is a concatenation of three different sources. First is the official WMT dev set that corresponds to the training data: newstest2018 for EN–DE, newstest-2013 for EN–FR, and newsdev2009 for EN–HU. To this we add the EUR-Lex and glossary dev sets described above. Tags are removed from these sets when they are used in the baseline system, which is not trained on any tagged data. Test sets are kept separated by data source. In addition to tagged and detagged versions of the EUR-Lex and glossary test sets described above, we include the EUR-Lex mono test set and the WMT official test sets: newstest2019 for EN–DE, newstest2014 for EN–FR, and newstest2009 for EN–HU. See Appendix A for the complete itemization of dev and test sets.

5.3 MT Systems

All our experiments are carried out using the Sockeye neural MT toolkit (Hieber et al., 2017). We use the Transformer architecture (Vaswani et al., 2017), with a hidden layer size of 512, an encoder of 20 layers, and a decoder of 2 layers: Hieber et al. (2020) report improved WMT results for such a configuration. For training, we set the batch size to 8192 tokens and the checkpoint interval to 2000 batches. Optimization is carried out with Sockeye's implementation of the Adam algorithm (Kingma and Ba, 2014). The learning rate, from an initial value of 0.0002, is multiplied by a factor of 0.9 every time eight training checkpoints pass without any improvement in dev-set perplexity. Training is stopped when there is no improvement after 32 checkpoints. Following convergence, the parameters from the eight best checkpoints are averaged.

We present a total of 33 experimental configurations: the detag-and-project baseline, plus the cross product of {1, 2, 5, 10, 15}% tag-injected data augmentation with {masked, raw} tag representation, for each of our three language pairs.

6 Results

6.1 Evaluation Metrics

Depending on the test set, we use a variety of evaluation metrics to judge performance.

We compute case-sensitive BLEU scores according to the SacreBLEU implementation (Post, 2018).[7] We distinguish untagged BLEU scores computed on test sets with no source-side tags (or for which the source-side tags have been removed) from tagged BLEU scores, where the tags are tokenized and treated as content by the built-in SacreBLEU tokenizer. Evaluation occurs only after any masked placeholders have been converted back to literal output. For each system, we also compute statistical significance relative to the baseline using stratified approximate randomization (Yeh, 2000).

Tagged test sets are also evaluated according to specifically designed "flagrant failure" metrics, whose goal is to detect obviously erroneous tag placement in MT output. Automatic evaluation of tag placement becomes difficult as the MT output diverges from a tagged reference translation's word choice, sentence structure, etc. Still, certain errors can be reliably detected regardless of language or context. We define flagrant-failure metrics to count occurrences of dropped, added, or mutilated tags, along with tags that become improperly nested relative to the source. In XLIFF, we distinguish a change of index — from e.g. <g id="2"> to <g id="4"> — as its own type of failure, rather than as independent drop and add mistakes.

Finally, we conduct a full human evaluation of tag placement accuracy, similar to the one introduced in Section 3.2. Due to the large amount of data involved, in this case we collect judgements from only a single annotator. Each tag is evaluated independently for whether its placement in the target is correct, incorrect, impossible given the MT output, missing, duplicated, or unclear.

6.2 Evaluation of Untagged Input

Our first concern is to ensure that augmenting the training data with tag-injected content does not harm translation of untagged inputs. We validate this claim on untagged versions of our WMT, EUR-Lex, and glossary test sets, comparing the BLEU scores of the tag-augmented systems with the score of the baseline, which was trained without tags.

[7] BLEU+case.mixed+numrefs.1+smooth.exp+tok.13a+-version.1.4.3

Lang	System	Set	ΔBLEU	*p*
EN–FR	Raw 10%	WMT	–0.4	0.02
EN–HU	Mask 15%	WMT	–0.3	0.03
EN–HU	Mask 5%	WMT	–0.3	0.04
EN–HU	Raw 10%	WMT	–0.3	0.05

Table 2: Experimental conditions with the most significant untagged BLEU differences relative to the baseline. Tag representation has no consistent effect on translation quality of untagged content.

	`<g id="2">`
Raw 1%	183,475
Raw 15%	2,606,016

	`<g id="@@ 3">`	
Raw 1%	183,475	3,150	1,300
Raw 15%	2,606,016	2,550	1,300

Figure 3: Training-data token frequencies for the BPE pieces involved in EN–FR translation of two tags. Indexes above 2 are seldom seen.

As hoped, we observe no clear pattern of degradation by tag representation, augmentation percentage, or language pair. Out of the 90 experimental cases, only four reach statistical difference with the corresponding baseline at $p < 0.05$. These scattered instances are shown in Table 2. See Appendix B for the full results.

6.3 Automatic Evaluation of Tag Placement

We have two proxy metrics at our disposal for automatically evaluating tag placement accuracy: tagged BLEU scores and flagrant-failure counts.

The previous section indicated that training-data augmentation did not have a strong effect on translation quality of *content*. We now turn to tagged BLEU scores to see if they provide any signal as to the translation quality of *markup*. As shown in Table 3, we observe a small positive improvement for the masking approach: statistical difference in 17 of 30 cases, and +0.79 BLEU on average when compared to the detag-and-project baseline. This effect is strongest for French and for 15% augmentation. At 15% masked augmentation, four of six results achieve statistical significance, and the average improvement is 1.00 BLEU. The best results for raw XLIFF systems appear scattered: the highest BLEU gains at 10% augmentation, the most consistent at 2%, and statistical significance achieved at least once everywhere except at 5% augmentation and in EN–DE.

We cross-check these conclusions by examining the flagrant failure rates. This analysis shows an extreme variability by test set. No configuration reaches more than a 3.0% failure rate on the EUR-Lex test set or 5.2% on the glossary test set. However, the increased diversity of tag indexes and tag patterns in the EUR-Lex mono test set provides for a much higher rate: up to 24.7% in the worst case. The full count of flagrant failures on the mono test set is displayed in Table 4.

By hard-coded design, the detag-and-project approach is not capable of changing, dropping, mutilating, or adding tags — they are placed without modification as a post-process after translation. We find that this technique also does not commit any flagrant errors of tag nesting on our test sets. Similar hard-coded limits affect the masking approach; the one instance of a dropped tag that we recorded is due to a tokenization error. Masking does, however, commit a certain number of nesting mistakes.[8] The raw-tag approach is susceptible to all kinds of flagrant failures. We note that, surprisingly, the number of errors tends to *increase* as more tagged examples are added to the training data in German and French, while Hungarian (with much smaller training data) does not show a clear pattern in any direction.

Increased errors in German are primarily due to more tags being generated with incorrect `id` parameters. Recall that our settings for tag injection (Section 5.1) introduce no more than two tag pairs into any sentence pair, resulting in a maximum `id` value of 2. The only examples of indexes beyond 2 in the training data come from the addition of tag self-translations (Section 4), which we include for indexes up to 20. These higher indexes never occur in the context of any content, and their relative prominence in the training data decreases as more tag-injected data is added, so the MT system may become less and less sure how to "translate" them in practice. The failures for German confirm this pattern. At 1% augmentation, all 30 tags with their IDs incorrectly changed have values beyond 2, but the MT system produces a different value beyond 2 in five cases. At 15% augmentation, the system does not propose a value beyond 2 in any of the 206 failure cases.

Arguably, the increased training focus on low-

[8] This count would include tags natively dropped by the MT system but re-added to the end of the output by rule. Masked systems produce on average 3% more line-final tags than raw systems, but essentially the same number as the baseline.

Language	Set	Detag/ Project	Masked 1%	2%	5%	10%	15%	Raw 1%	2%	5%	10%	15%
EN–DE	EUR-Lex	70.0	0.1	0.2	0.4	0.1	0.3	–0.2	0.2	0.2	–0.1	0.2
	Glossary	60.2	1.1	0.6	1.2	0.8	0.8	1.0	0.6	1.0	0.8	0.8
EN–FR	EUR-Lex	65.9	1.0	0.1	0.9	0.1	1.2	0.5	0.5	0.3	0.9	–0.5
	Glossary	61.7	0.9	0.9	0.9	1.0	1.7	1.1	0.6	0.7	0.7	0.2
EN–HU	EUR-Lex	61.4	0.2	0.3	0.2	0.4	0.6	–0.5	0.0	–0.1	–0.2	–0.1
	Glossary	53.5	1.2	1.6	1.4	1.8	1.5	–0.3	0.3	0.6	1.8	1.3

Table 3: BLEU scores on tagged test sets, shown as differences from the baseline (detag-and-project) system's performance on the same test set. Results in grey are statistically significant at $p < 0.05$. The masked representation tends to produce the best results.

Language	Failure	DP	Masked 1%	2%	5%	10%	15%	Raw 1%	2%	5%	10%	15%
EN–DE	Changed ID	0	0	0	0	0	0	30	66	177	226	206
	Dropped	0	0	0	0	0	0	92	105	128	162	131
	Mutilated	0	0	0	0	0	0	17	3	85	91	129
	Badly Nested	0	12	14	14	15	10	26	25	33	28	31
	Added	0	0	0	0	0	0	22	11	14	10	6
	Total	0	12	14	14	15	10	187	210	437	517	503
EN–FR	Changed ID	0	0	0	0	0	0	5	112	325	349	95
	Dropped	0	0	0	0	0	0	34	81	82	115	423
	Mutilated	0	0	0	0	0	0	1	59	232	249	257
	Badly Nested	0	9	16	9	20	16	10	16	16	28	162
	Added	0	0	0	0	0	0	10	21	17	18	36
	Total	0	9	16	9	20	16	60	289	672	759	973
EN–HU	Changed ID	0	0	0	0	0	0	337	357	319	358	306
	Dropped	0	0	1	1	0	1	265	243	257	221	272
	Mutilated	0	0	0	0	0	0	24	19	57	4	45
	Badly Nested	0	26	10	8	16	17	39	28	35	37	33
	Added	0	0	0	0	0	0	37	44	43	28	18
	Total	0	26	11	9	16	18	702	691	711	648	674

Table 4: Flagrant failure counts on the EUR-Lex mono tagged test set. ("DP" = detag and project.) Tag translation failures increase rapidly as the training data is augmented with more raw tags.

index tags should be equally true of the mask-based systems. A key difference is illustrated by the rise in mutilated tags for French. Masked tags are always expressed as single tokens; if the self-translated examples are enough to induce a copy-through behavior for them, the behavior can be correctly applied in any content segment regardless of context. However, raw tags are expressed as at least two and sometimes more tokens (cf. Figure 2), which turn out to have wildly different frequencies in the training data as augmentation increases.

Figure 3 illustrates the BPE pieces involved in translating the raw tags `<g id="2">` and `<g id="3">` from English to French, along with the observed frequencies of those tokens in the training data. For a low-index tag, the tokens are quite common and, moreover, have relatively balanced counts: the opening token is seen roughly 2.4 times as often as the closing token no matter the data augmentation percentage. The situation is markedly different for a higher-index tag. In this case, increasing amounts of tag injection heavily shift vocabulary mass toward the opening token: it already appears 58 times more frequently than the tail at 1% augmentation, a ratio that grows to 1022 at 15%.

This extreme imbalance may be responsible for the 15% model's tag mutilation behavior. On our mono test set, this model never produces the first token of a tag without the tail, nor does it mutilate any tags for indexes up to 2. All the mutilation failures are caused by producing the tail alone for higher indexes, e.g. `id="3">` — exactly corresponding to the string of low-frequency tokens where a copy-through behavior can be easily learned.

6.4 Human Evaluation of Tag Placement

We conduct a human evaluation of tag placement accuracy in order to get a more complete picture of errors than afforded by tagged BLEU scores and flagrant-failure counts. Since it is impractical to

(a) Lines with indexes 1 and 2 only

Lang	System	Good	Bad	Residual
EN–DE	detag/project	97.3%	2.7%	0.0%
	masked (15%)	98.6%	1.4%	0.0%
	raw (1%)	98.4%	1.6%	0.0%
EN–FR	detag/project	92.9%	4.0%	3.1%
	masked (15%)	97.0%	0.5%	2.5%
	raw (1%)	96.1%	1.4%	2.5%
EN–HU	detag/project	93.8%	4.9%	1.3%
	masked (15%)	96.9%	2.5%	0.9%
	raw (10%)	96.1%	3.0%	0.9%

(b) Lines with indexes 3 and above

Lang	System	Good	Bad	Residual
EN–DE	detag/project	87.3%	12.7%	0.0%
	masked (15%)	84.2%	15.8%	0.0%
	raw (1%)	83.6%	16.0%	0.4%
EN–FR	detag/project	77.7%	9.8%	12.5%
	masked (15%)	78.7%	9.0%	12.3%
	raw (1%)	80.8%	6.2%	12.9%
EN–HU	detag/project	77.9%	19.3%	2.7%
	masked (15%)	70.9%	19.5%	9.6%
	raw (10%)	33.0%	63.8%	3.2%

Table 5: Summarized human evaluation of tag placement accuracy.

collect judgements on full test sets for all 33 experiments, we restrict this study to subsets of both. In terms of systems, we evaluate the detag-and-project baseline, the 15% masked augmentation, and either the 1% raw (EN–DE, EN–FR) or 10% raw (EN–HU) augmentation. For data, we use the entire glossary test set (289 lines), all the lines in the EUR-Lex monolingual test set containing tag indexes 3 and above (283 lines), and an equal number of randomly sampled lines from the same test set containing indexes 1 and 2 only.

In summarizing the results, we collapse the eight annotation types into three categories. "Good" tags are those placed correctly in the output, including if they were correctly deleted or duplicated. "Bad" tags are incorrectly placed, incorrectly dropped or duplicated, hallucinated, or mutilated in the output. "Residual" tags are those judged as impossible or unclear to place. Given the marked difference in flagrant failures observed for tag indexes 1 and 2 versus 3 and beyond, we report human judgements separately by whether the input included indexes beyond 2 or not. These summarized results appear in Table 5.

Placement for low-index tags present in the augmented training data is learned quite well: in all language pairs, the masked and raw-tag systems outperform the detag-and-project baseline. Humans also find the annotation of inputs with few tags to be a straightforward task, as very few tags are marked as awkward to place.

Results are less clear-cut on high-index tags that appear in training only as self-translated examples. Word-alignment-based projection works best in EN–DE and EN–HU. Translating raw tags does well in EN–DE and EN–FR but is unusable in EN–HU. The masking approach performs consistently in second place. Especially in French, the human task of judging placement accuracy has become notably harder, an effect that could also significantly affect the good/bad results of any method.

7 Conclusion

We have performed a comprehensive evaluation of several tag representation methods and proposed a data-augmentation technique that allows MT models to jointly learn content translation and inline tag placement.

Results show that representing tags as masks, together with data augmentation, leads to equivalent or improved performance over a detag-and-project approach: placement accuracy is higher for tags frequent in the training data, while it may vary for tags never observed in context. In practice, it may be preferable to rely on the MT model's ability to learn mask placement — even with some variability in accuracy — than to implement, debug, and maintain the baseline's more complicated projection rules and the required alignment model.

Raw tags, on the other hand, fail our generalization tests. Though placement accuracy is again baseline-beating for commonly observed tags, raw models seem unable to copy rare tags into the output without a significant number of mutilations, deletions, and duplications: an unacceptable result for the goal of obtaining well-structured output.

Several changes to our setup may improve the transfer of raw tags. Injecting XLIFF tags with a wider variety of `id` values is needed to expose the model to them in context instead of merely in self-translation. Explicitly identifying tag tokens via input factors (Dinu et al., 2019), or constraining/promoting the output of complete tags (Hashimoto et al., 2019), would also be helpful for reducing the rate of malformed output.

Acknowledgements

We thank Yaser Al-Onaizan, Marcello Federico, Stanislas Lauly, and Prashant Mathur for early discussions regarding the tag-injection technique.

References

Roee Aharoni and Yoav Goldberg. 2017. Towards string-to-tree neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 132–140, Vancouver, Canada. Association for Computational Linguistics.

Alexandre Bérard, Ioan Calapodescu, and Claude Roux. 2019. Naver Labs Europe's systems for the WMT19 machine translation robustness task. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 526–532, Florence, Italy. Association for Computational Linguistics.

Alexandra Birch, Barry Haddow, Ivan Tito, Antonio Valerio Miceli Barone, Rachel Bawden, Felipe Sánchez-Martínez, Mikel L. Forcada, Miquel Esplà-Gomis, Víctor Sánchez-Cartagena, Juan Antonio Pérez-Ortiz, Wilker Aziz, Andrew Secker, and Peggy van der Kreeft. 2019. Global under-resourced media translation (GoURMET). In *Proceedings of Machine Translation Summit XVII Volume 2: Translator, Project and User Tracks*, pages 122–122, Dublin, Ireland. European Association for Machine Translation.

Josep Crego, Jungi Kim, Guillaume Klein, Anabel Rebollo, Kathy Yang, Jean Senellart, Egor Akhanov, Patrice Brunelle, Aurelien Coquard, Yongchao Deng, Satoshi Enoue, Chiyo Geiss, Joshua Johanson, Ardas Khalsa, Raoum Khiari, Byeongil Ko, Catherine Kobus, Jean Lorieux, Leidiana Martins, Dang-Chuan Nguyen, Alexandra Priori, Thomas Riccardi, Natalia Segal, Christophe Servan, Cyril Tiquet, Bo Wang, Jin Yang, Dakun Zhang, Jing Zhou, and Peter Zoldan. 2016. SYSTRAN's pure neural machine translation systems. *Computing Research Repository*, arXiv:1610.05540. Version 1.

Georgiana Dinu, Prashant Mathur, Marcello Federico, and Yaser Al-Onaizan. 2019. Training neural machine translation to apply terminology constraints. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3063–3068, Florence, Italy. Association for Computational Linguistics.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM Model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648, Atlanta, Georgia. Association for Computational Linguistics.

Marcello Federico, Nicola Bertoldi, Mauro Cettolo, Matteo Negri, Marco Turchi, Marco Trombetti, Alessandro Cattelan, Antonio Farina, Domenico Lupinetti, Andrea Martines, Alberto Massidda, Holger Schwenk, Loïc Barrault, Frederic Blain, Philipp Koehn, Christian Buck, and Ulrich Germann. 2014. The MateCat tool. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: System Demonstrations*, pages 129–132, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Kazuma Hashimoto, Raffaella Buschiazzo, James Bradbury, Teresa Marshall, Richard Socher, and Caiming Xiong. 2019. A high-quality multilingual dataset for structured documentation translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 116–127, Florence, Italy. Association for Computational Linguistics.

Felix Hieber, Tobias Domhan, Michael Denkowski, and David Vilar. 2020. Sockeye 2: A toolkit for neural machine translation. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 457–458, Lisbon, Portugal.

Felix Hieber, Tobias Domhan, Michael Denkowski, David Vilar, Artem Sokolov, Ann Clifton, and Matt Post. 2017. Sockeye: A toolkit for neural machine translation. *Computing Research Repository*, arXiv:1712.05690. Version 2.

Eric Joanis, Darlene Stewart, Samuel Larkin, and Roland Kuhn. 2013. Transferring markup tags in statistical machine translation: A two-stream approach. In *Proceedings of MT Summit XIV Workshop on Post-editing Technology and Practice*, pages 73–81, Nice, France.

Vladimir Karpukhin, Omer Levy, Jacob Eisenstein, and Marjan Ghazvininejad. 2019. Training on synthetic noise improves robustness to natural noise in machine translation. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 42–47, Hong Kong, China. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *Computing Research Repository*, arXiv:1412.6980. Version 9.

Kasia Kosmaczewska and Matt Train. 2019. Application of post-edited machine translation in fashion eCommerce. In *Proceedings of Machine Translation Summit XVII Volume 2: Translator, Project and User Tracks*, pages 167–173, Dublin, Ireland. European Association for Machine Translation.

Paul Michel and Graham Neubig. 2018. MTNT: A testbed for machine translation of noisy text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 543–553, Brussels, Belgium. Association for Computational Linguistics.

Mathias Müller. 2017. Treatment of markup in statistical machine translation. In *Proceedings of the Third Workshop on Discourse in Machine Translation*, pages 36–46, Copenhagen, Denmark. Association for Computational Linguistics.

Soichiro Murakami, Makoto Morishita, Tsutomu Hirao, and Masaaki Nagata. 2019. NTT's machine translation systems for WMT19 robustness task. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 544–551, Florence, Italy. Association for Computational Linguistics.

Maria Nădejde, Siva Reddy, Rico Sennrich, Tomasz Dwojak, Marcin Junczys-Dowmunt, Philipp Koehn, and Alexandra Birch. 2017. Predicting target language CCG supertags improves neural machine translation. In *Proceedings of the Second Conference on Machine Translation*, pages 68–79, Copenhagen, Denmark. Association for Computational Linguistics.

Mara Nunziatini. 2019. Machine translation in the financial services industry: A case study. In *Proceedings of Machine Translation Summit XVII Volume 2: Translator, Project and User Tracks*, pages 57–63, Dublin, Ireland. European Association for Machine Translation.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Matt Post, Shuoyang Ding, Marianna Martindale, and Winston Wu. 2019. An exploration of placeholding in neural machine translation. In *Proceedings of Machine Translation Summit XVII Volume 1: Research Track*, pages 182–192, Dublin, Ireland. European Association for Machine Translation.

Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán. 2019. WikiMatrix: Mining 135M parallel sentences in 1620 language pairs from Wikipedia. *Computing Research Repository*, arXiv:1907.05791. Version 2.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Rico Sennrich and Barry Haddow. 2016. Linguistic input features improve neural machine translation. In *Proceedings of the First Conference on Machine Translation: Volume 1, Research Papers*, pages 83–91, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Alexander Yeh. 2000. More accurate tests for the statistical significance of result differences. In *COLING 2000 Volume 2: The 18th International Conference on Computational Linguistics*.

A Train, Dev, and Test Corpora

Table 6 gives a more detailed account of our training resources, showing the amount of material sourced from each standard WMT corpora we use. Following Figure 2 of Schwenk et al. (2019), we reduce the EN–DE WikiMatrix corpus to only those lines with a margin threshold of 1.05 or higher *and* where the WMT-provided language detection registered English on the source and German on the target. This is in addition to ignoring Common Crawl and Paracrawl entirely. For EN–FR, we discard the Common Crawl and Giga-FrEn corpora.

Clean-up on the training data is limited to removing lines where the source and/or target side is blank, removing lines whose source side is contained in one of our dev or test sets, tokenization, byte-pair encoding (Sennrich et al., 2016) using 32,000 operations, and length-based filtering. In this final step, we remove sentence pairs of more than 95 tokens on either side, containing tokens with more than 100 characters, or where the length ratio between source and target is too unbalanced.

Language	Corpus	Lines
EN–DE	Europarl	1,828,521
	News Commentary	371,225
	Rapid	1,631,639
	WikiMatrix (filtered)	916,242
	WikiTitles	1,382,687
	Total	6,130,314
	After filtering	5,746,433
EN–FR	Europarl	2,007,723
	News Commentary	183,251
	UN Docs	12,886,831
	Total	15,077,805
	After filtering	14,467,303
EN–HU	Hung-Train	1,517,584
	Total	1,517,584
	After filtering	1,465,919

Table 6: Line counts of baseline training data.

Language	Set	Source	Lines
EN–DE	Dev	WMT (nt2018)	2,998
		EUR-Lex	1,888
		Glossary	286
	Test	WMT (nt2019)	1,997
		EUR-Lex	1,450
		EUR-Lex mono	2,525
		Glossary	289
EN–FR	Dev	WMT (nt2013)	3,000
		EUR-Lex	1,888
		Glossary	286
	Test	WMT (nt2014)	3,003
		EUR-Lex	1,450
		EUR-Lex mono	2,525
		Glossary	289
EN–HU	Dev	WMT (nd2009)	2,051
		EUR-Lex	1,888
		Glossary	286
	Test	WMT (nt2009)	3,027
		EUR-Lex	1,450
		EUR-Lex mono	2,525
		Glossary	289

Table 7: Line counts of the dev and test sets.

Table 7 shows the sizes of our final dev and test sets, including WMT "newsdev" (nd) and "newstest" (nt) releases.

B Additional Results

Table 8 (on the next page) shows complete results on translating *untagged* test sets, to ensure that adding masked or raw tags to our training data does not adversely affect the translation of plain content. BLEU scores are computed according to SacreBLEU (Post, 2018), while statistical significance uses 1000 trials of stratified approximate randomization (Yeh, 2000). The small glossary test set shows the highest BLEU variance, but only once to statistical significance. Meanwhile, the few significant differences are scattered across language pairs, test sets, tag representations, and augmentation percentages.

			Masked					Raw				
Language	**Set**	**Baseline**	**1%**	**2%**	**5%**	**10%**	**15%**	**1%**	**2%**	**5%**	**10%**	**15%**
EN–DE	WMT	38.6	0.1	0.3	0.3	–0.1	0.0	0.1	0.2	0.1	–0.2	–0.3
	EUR-Lex	44.4	–0.5	–0.1	0.6	–0.3	–0.4	–0.9	0.5	0.3	0.1	0.1
	Glossary	39.9	0.6	–0.2	0.0	1.0	1.0	0.5	–0.5	0.2	0.0	0.5
EN–FR	WMT	37.5	–0.2	–0.1	–0.1	–0.1	0.2	0.0	–0.2	–0.1	–0.4	0.0
	EUR-Lex	43.0	–0.1	–0.3	0.4	–0.1	0.5	0.5	–0.1	0.0	0.5	–0.3
	Glossary	45.7	0.3	–0.2	0.1	0.3	0.9	0.7	0.7	0.3	0.7	–0.2
EN–HU	WMT	12.9	0.0	–0.1	–0.3	0.1	–0.3	0.0	0.0	–0.3	–0.3	0.0
	EUR-Lex	27.6	–0.4	–0.3	–0.5	–0.2	0.3	–0.2	0.1	–0.6	–0.1	0.0
	Glossary	27.4	–0.2	–0.3	–1.0	0.4	–0.7	–0.8	–0.4	–0.6	1.1	–0.1

Table 8: BLEU scores on untagged test sets, shown as differences from the baseline system's performance on the same test set. Cells in light grey are statistically significant at $p < 0.10$; dark grey indicates $p < 0.05$. Tag representation has no consistent effect on translation quality of untagged content.

The Tatoeba Translation Challenge – Realistic Data Sets for Low Resource and Multilingual MT

Jörg Tiedemann
University of Helsinki
jorg.tiedemann@helsinki.fi
https://github.com/Helsinki-NLP/Tatoeba-Challenge

Abstract

This paper describes the development of a new benchmark for machine translation that provides training and test data for thousands of language pairs covering over 500 languages and tools for creating state-of-the-art translation models from that collection. The main goal is to trigger the development of open translation tools and models with a much broader coverage of the World's languages. Using the package it is possible to work on realistic low-resource scenarios avoiding artificially reduced setups that are common when demonstrating zero-shot or few-shot learning. For the first time, this package provides a comprehensive collection of diverse data sets in hundreds of languages with systematic language and script annotation and data splits to extend the narrow coverage of existing benchmarks. Together with the data release, we also provide a growing number of pre-trained baseline models for individual language pairs and selected language groups.

1 Introduction

The Tatoeba translation challenge includes shuffled training data taken from OPUS,[1] an open collection of parallel corpora (Tiedemann, 2012), and test data from Tatoeba,[2] a crowd-sourced collection of user-provided translations in a large number of languages. All data sets are labeled with ISO-639-3 language codes using macro-languages in case when available. Naturally, training data do not include sentences from Tatoeba and neither from the popular WMT testsets to allow a fair comparison to other models that have been evaluated using those data sets.

Here, we propose an open challenge and the idea is to encourage people to develop machine translation in real-world cases for many languages. The most important point is to get away from artificial setups that only simulate low-resource scenarios or zero-shot translations. A lot of research is tested with multi-parallel data sets and high resource languages using data sets such as WIT^3 (Cettolo et al., 2012) or Europarl (Koehn, 2005) simply reducing or taking away one language pair for arguing about the capabilities of learning translation with little or without explicit training data for the language pair in question (see, e.g., Firat et al. (2016a,b); Ha et al. (2016); Lakew et al. (2018)). Such a setup is, however, not realistic and most probably over-estimates the ability of transfer learning making claims that do not necessarily carry over towards real-world tasks.

In the set we provide here we, instead, include all available data from the collection without removing anything. In this way, the data refers to a diverse and skewed collection, which reflects the real situation we need to work with and many low-resource languages are only represented by noisy or very unrelated training data. Zero-shot scenarios are only tested if no data is available in any of the sub-corpora. More details about the data compilation and releases will be given below.

Tatoeba is, admittedly, a rather easy test set in general but it includes a wide variety of languages and makes it easy to get started with rather encouraging results even for lesser resourced languages. The release also includes medium and high resource settings and allows a wide range of experiments with all supported language pairs including studies of transfer learning and pivot-based methods.

2 Data releases

The current release includes over 500GB of compressed data for 2,961 language pairs covering 555 languages. The data sets are released per language

[1] http://opus.nlpl.eu/
[2] https://tatoeba.org/

Proceedings of the 5th Conference on Machine Translation (WMT), pages 1174–1182
Online, November 19–20, 2020.

pair with the following structure, using deu-eng as an example (see Figure 1).

```
data/deu-eng/
data/deu-eng/train.src.gz
data/deu-eng/train.trg.gz
data/deu-eng/train.id.gz
data/deu-eng/dev.id
data/deu-eng/dev.src
data/deu-eng/dev.trg
data/deu-eng/test.src
data/deu-eng/test.trg
data/deu-eng/test.id
```

Figure 1: Released data packages: training data, development data and test data. Language labels are stored in ID files that also contain the name of the source corpus for the training data sets.

Files with the extension *.src* refer to sentences in the source language (*deu* in this case) and files with extension *.trg* contain sentences in the target language (*eng* here). File with extension *.id* include the ISO-639-3 language labels with possibly extensions about the orthographic script (more information below). In the *.id* file for the training data there are also labels for the OPUS corpus the sentences come from. We include the entire collection available from OPUS with data from the following corpora: ada83, Bianet, bible-uedin, Books, CAPES, DGT, DOGC, ECB, EhuHac, EiTB-ParCC, Elhuyar, EMEA, EUbookshop, EUconst, Europarl, Finlex, fiskmo, giga-fren, GlobalVoices, GNOME, hrenWaC, infopankki, JRC-Acquis, JW300, KDE4, KDEdoc, komi, MBS, memat, MontenegrinSubs, MultiParaCrawl, MultiUN, News-Commentary, OfisPublik, OpenOffice, OpenSubtitles, ParaCrawl, PHP, QED, RF, sardware, SciELO, SETIMES, SPC, Tanzil, TED2013, TedTalks, TEP, TildeMODEL, Ubuntu, UN, UNPC, wikimedia, Wikipedia, WikiSource, XhosaNavy.

The data sets are compiled from the pre-aligned bitexts but further cleaned in various ways. First of all, we remove non-printable characters and strings that violate Unicode encoding principles using regular expressions and a recoding trick using the forced encoding mode of *recode* (v3.7), a popular character conversion tool.[3] Furthermore, we also de-escape special characters (like '&' encoded as '&') that may appear in some of the corpora. For that, we apply the tools from Moses (Koehn et al., 2007). Finally, we also apply automatic language identification to remove additional noise from the data. We use the compact language detect library (CLD2) through its Python bindings[4] and a Python library for converting between different ISO-639 standards.[5] CLD2 supports 172 languages and we use the options for "best effort" and apply the assumed language from the original data as the "hint language code". For unsupported languages, we remove all examples that are detected to be English as this is a common problem in some corpora where English texts appear in various places (e.g. untranslated text in localization data of community efforts). In all cases, we only rely on the detected language if it is flagged as reliable by the software.

All corpus data and sub-languages are merged and shuffled using terashuf[6] that is capable to efficiently shuffle large data sets. But we keep track of the original data set and provide labels to recognize the origin. In this way, it is possible to restrict training to specific subsets of the data to improve domain match or to reduce noise. The entire procedure of compiling the Tatoeba Challenge data sets is available from the project repository at https://github.com/Helsinki-NLP/Tatoeba-Challenge.

The largest data set (English-French) contains over 180 million aligned sentence pairs and 173 language pairs are covered by over 10 million sentence pairs in our collection. Altogether, there are almost bilingual 3,000 data sets and we plan regular updates to improve the coverage. Below, we give some more details about the language labels, test sets and monolingual data sets that we include in the package as well.

2.1 Language labels and scripts

We label all data sets with standardized language codes using three-letter codes from ISO-639-3. The labels are converted from the original OPUS language IDs (which roughly follow ISO-639-1 codes but also include various non-standard IDs) and information about the writing system (or script) is automatically assigned using Unicode regular expressions and counting letters from specific script character properties. For the scripts we use four-letter codes from ISO-15924 and attach them to the three-letter language codes defined in ISO-639-3. Only the most frequently present script in a string is shown. Mixed content may appear but is not marked specifically. Note that the code Zyyy

[3] https://github.com/pinard/Recode

[4] https://pypi.org/project/pycld2/

[5] https://pypi.org/project/iso-639/

[6] https://github.com/alexandres/terashuf

refers to common characters that cannot be used to distinguish scripts. The information about the script is not added if there is only one script in that language and no other scripts are detected in any of the strings. If there is a default script among several alternatives then this particular script is not shown either. Note that the assignment is done fully automatically and no corrections have been made. Three example label sets are given below using the macro-languages Chinese (zho), Serbo-Croatian (hbs) and Japanese (jpn) that can use character from different scripts:

Chinese: cjy_Hans, cjy_Hant, cmn, cmn_Bopo, cmn_Hans, cmn_Hant, cmn_Latn, gan, lzh, lzh_Bopo, lzh_Hang, lzh_Hani, lzh_Hans, lzh_Hira, lzh_Kana, lzh_Yiii, nan_Hani, nan_Latn, wuu, wuu_Bopo, wuu_Hang, wuu_Hani, wuu_Hira, yue_Hans, yue_Hant, yue_Latn

Japanese: jpn, jpn_Hani, jpn_Hira, jpn_Kana, jpn_Latn

Serbo-Croatian: bos_Latn, hrv, srp_Cyrl, srp_Latn

This demonstrates that a data set may include examples from various sub-languages if they exist (e.g. Bosnian, Croatian and Serbian in the Serbo-Croatian case) or language IDs with script extensions that show the dominating script in the corresponding string (e.g. Cyrl for Cyrillic or Latn for Latin script). Those labels can be used to separate the data sets, to test sub-languages or specific scripts only or to remove some noise (like the examples that are tagged with the Latin script (Latn) in the Japanese data set. Note that script detection can also fail in which the corresponding code is missing or potentially wrong. For example, the detection of traditional (Hant) och simplified Chinese (Hans) can be ambiguous and encoding noise can have an effect on the detection.

We also release the tools that we developed for converting and standardizing OPUS IDs and also the tools that detect scripts and variants of writing systems. The package is available from github[7] and can be installed from CPAN.[8]

2.2 Multiple reference translations

Test and development data are taken from a shuffled version of Tatoeba. All translation alternatives are included in the data set to obtain the best coverage of languages in the collection. Development and test sets are disjoint in the sense that they do not include identical source-target language sentence pairs. However, there can be identical source sentences or identical target sentences in both sets, which are not linked to the same translations. Similarly, there can be identical source or target sentences in one of the sets, for example the test set, with different translations. In Figure 2, you can see examples from the Esperanto-Ladino test set.

epo	lad_Latn
u vi estas en Berlino?	Estash en Berlin?
u vi estas en Berlino?	Vos estash en Berlin?
u vi estas en Berlino?	Vozotras estash en Berlin?
La hundo estas nigra.	El perro es preto.
La hundo nigras.	El perro es preto.

Figure 2: Examples of test sentences with multiple reference translations taken from the Esperanto-Ladino test set.

The test data could have been organized as multi-reference data sets but this would require to provide different sets in both translation directions. Removing alternative translations is also not a good option as this would take away a lot of relevant data. Hence, we decided to provide the data sets as they are, which implicitly creates multi-reference test sets but with the wrong normalization.

2.3 Monolingual data

In addition to the parallel data sets we also provide monolingual data that can be used for unsupervised methods or data augmentation approaches such as back-translation. For that purpose, we extract public data from Wikimedia including source from Wikpedia, Wikibooks, Wikinews, Wikiquote and Wikisource. We extract sentences from data dumps provided in JSON format[9] and process them with jq,[10] a lightweight JSON processing tool. We apply the same cleaning steps as we do for the OPUS bitexts including language identification and convert language IDs to ISO-639-3 as before. Sentence boundaries are detected using UDPipe (Straka et al., 2016) with models trained on universal dependency treebanks v 2.4 and the Moses sentence splitter with language-specific non-breaking prefixes if available. We preserve document boundaries and do not shuffle the data to enable experiments with discourse-aware models. The data sets are released along with the rest of the Tatoeba challenge data.

3 The translation challenge

The main challenge is to develop translation models and to test them with the given test data from

[7] https://github.com/Helsinki-NLP/LanguageCodes

[8] https://metacpan.org/pod/ISO::639::3 and https://metacpan.org/pod/ISO::639::5

[9] https://dumps.wikimedia.org/other/cirrussearch/current

[10] https://stedolan.github.io/jq/

Tatoeba. The focus is on low-resource languages and to push their coverage and translation quality. Resources for high-resource are also provided and can be used as well for translation modeling of those languages and for knowledge transfer to less resourced languages. Note that not all language pairs have sufficient data sets for test, development (dev) and training ($train$) data. Hence, we divided the Tatoeba challenge data into various subsets based on the size of the training data available.

high-resource settings: 298 language pairs with training data of at least one million training examples (aligned sentence pairs), we further split into language pairs with more than 10 million training examples (173 language pairs) and other language pairs with data sets below the size of 10 million examples

medium-sized resource settings: 97 language pairs with more than 100,000 and less than 1 million training examples

low-resource settings: 87 language pairs with less than 100,000 training examples, we further distinguish between language pairs with more than 10,000 training examples (63) and language pairs below 10,000 training examples (24)

zero-shot translation: language pairs with no training data (40 in the current data set)

For all those 522 selected language pairs, the data set provides at least 200 sentences per test set. 101 of them involves English as one of the languages. 288 test sets contain more than 1,000 sentence pairs of which only 68 include English. Note, that everything below 1,000 sentences is probably not very reliable as a proper test set but we decided to release smaller test sets as an initial benchmark to trigger further development even for extremely under-resourced language pairs. We also decided to use very low thresholds for the division into low-resource languages. Having 10,000 training examples or less is very realistic for many real-world examples and we want to encourage the work on such cases in particular.

The maximum size of test sets in our collection is 10,000 sentence pairs, which is available for 76 language pairs. The test size is reduced to 5,000 if there is less than 20,000 sentence pairs in Tatoeba (19 data sets). The remaining sentences are released as disjoint validation data. For 48 Tatoeba language pairs with less than 10,000 sentence pairs, we keep 2,500 for the test set and the rest for validation and for 78 Tateoba language pairs with less than 5,000 sentence pairs we keep 1,000 for validation and the rest for testing. Finally, for language pairs with less than 2,000 sentences in Tatoeba we skip validation data and use everything for test purposes.

Test and validation data are strictly disjoint and none of the examples from Tatoeba are explicitly included in the training data. However, as it is common in realistic cases, there is a natural chance for a certain overlap between those data sets. Figure 3 plots the percentage of sentence pairs in test and validation sets that can also be found in the corresponding training data we release. The average proportion is rather low around 5.5% for both with a median percentage of 2.3% and 2.9% for test and validation data, respectively. There is one clear outlier with a very high proportion of over 55% overlap and that is Danish–English for some reason that is not entirely clear to us. Otherwise, the values are well below that ratio.

4 The data challenge

The most important ingredient for improved translation quality is data. It is not only about training data but very much also about appropriate test data that can help to push the development of transfer models and other ideas of handling low-resource settings. Therefore, another challenge we want to open here is the increase of the coverage of test sets for low-resource languages. Our strategy is to organize the extension of the benchmarks directly through the Tatoeba initiative. Users who would like to contribute to further MT benchmark development are asked to register for the open service provided by Tatoeba and to upload new translations in the languages of interest. From our side, we will continuously update our challenge data set to include the latest data releases coming from Tatoeba including new language pairs and extended data sets for existing language pairs. We will make sure that the new test sets do not overlap with any released development data from previous revisions to enable fair comparisons of old models with new benchmarks. The extended test and validation data sets will be released as new packages and old revisions will be kept for replicability of existing

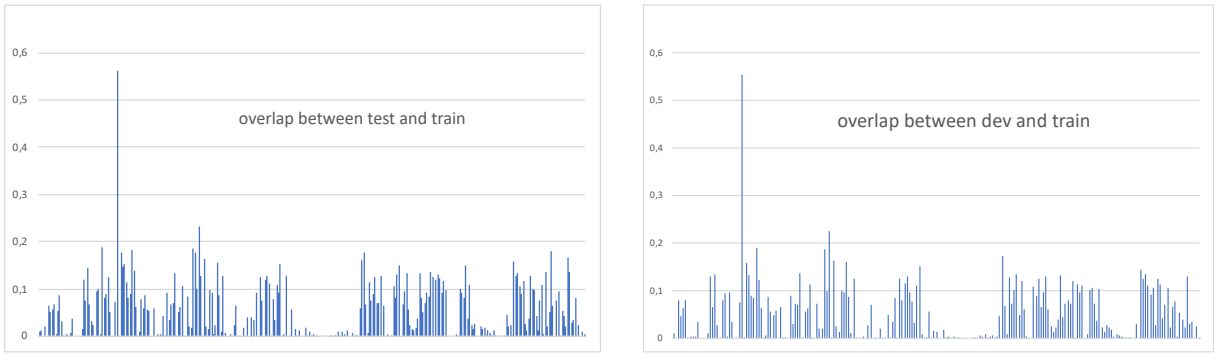


Figure 3: Overlap between test and validation (dev) data and the training data: Proportion of sentence pairs that exist in the training data for all data sets above 1,000 sentence pairs.

scores.

In order to provide information about language pairs in need, we provide a list of data sets with less than 1,000 examples per language pair. In the current release, this refers to 2,375 language pairs. 2,141 language pairs have less than 200 translation units and are, therefore, not included in the released benchmark test set. Furthermore, we also provide a list of languages for which we release training data coupled with English but no test data is available from Tatoeba. Currently, this relates to 246 languages.

We encourage users to especially contribute translations for those data sets in order to improve the language coverage even further. We hope to trigger a grass-root development that can significantly boost the availability of development and test sets as one of the crucial elements for pushing NMT development in the corresponding languages.

Finally, we also encourage to incorporate other test sets besides of the Tatoeba data. Currently, we also test with WMT news test sets for the language pairs that are covered by the released development and test sets over the years of the news translation campaign. Contributions and links can be provided through the repository management interface at github.

5 How to participate

The goal of the data release is to enable a straightforward setup for machine translation development. Everyone interested is free to use the data for their own development. A leader board for individual language pairs will be maintained. Furthermore, we also intend to make models available that are listed in the challenge. This does not only support replicability but also provides a new unique resource of pre-trained models that can be integrated in real-world applications or can be used in further research, unrelated downstream tasks or as a starting point for subsequent fine-tuning and domain adaptation. A large number of models is already available from our side providing baselines for a large portion of the data set. More details will be provided below.

For participation, there are certain rules that apply:

- Do not use any development or test data for training (dev can be used for validation during training as an early stopping criterion).
- Only use the provided training data for training models with comparable results in constrained settings. Any combination of language pairs is fine or backtranslation of sentences included in training data for any language pair is allowed, too. That means that additional data sets, parallel or monolingual, are not allowed for official models to be compared with others.
- Unconstrained models may also be trained and can be reported as a separate category. Using pre-trained language or translation models fall into the unconstrained category. Make sure that the pre-trained model does not include Tatoeba data that we reserve for testing.
- We encourage to release models openly to ensure replicability and re-use of pre-trained models. If you want to enter the official leader board you have to make your model available including instructions on how to use them.

6 Baseline Models

Along with the data, we also release baseline models that we train with state-of-the-art trans-

former models (Vaswani et al., 2017) using Marian-NMT,[11] a stable production-ready NMT toolbox with efficient training and decoding capabilities (Junczys-Dowmunt et al., 2018). We apply a common setup with 6 self-attentive layers in both, the encoder and decoder network using 8 attention heads in each layer. The hyper-parameters follow the general recommendations given in the documentation of the software.[12] The training procedures follow the strategy implemented in OPUS-MT (Tiedemann and Thottingal, 2020) and detailed instructions are available from github.[13]

We train a selection of models on v100 GPUs with early-stopping after 10 iterations of dropping validation perplexities. We use SentencePiece (Kudo and Richardson, 2018) for the segmentation into subword units and apply a shared vocabulary of a maximum of 65,000 items. Language label tokens in the spirit of Johnson et al. (2017) are used in case of multiple language variants or scripts in the target language. Models for over 400 language pairs are currently available and we refer the reader to the website with the latest results. For illustration, we provide some example scores below in Table 1 using automatic evaluation based on chrF2 and BLEU computed using sacrebleu (Post, 2018). The actual translations are also available for each model and the distribution comes along with the logfiles from the training process and all necessary data files such as the SentencePiece models and vocabularies.

language pair	chrF2	BLEU
aze-eng	0.490	31.9
bel-eng	0.268	10.0
cat-eng	0.668	50.2
eng-epo	0.577	35.6
eng-glg	0.593	37.8
eng-hye	0.404	16.6
eng-ilo	0.569	30.8
eng-run	0.436	10.4

Table 1: Translations scores from baseline models trained for a selection of medium-size language pairs (according to our classification) tested on the provided Tatoeba benchmark. We show here models that include English and score above 10 BLEU.

[11] https://marian-nmt.github.io

[12] https://github.com/marian-nmt/marian-examples/tree/master/transformer

[13] https://github.com/Helsinki-NLP/OPUS-MT-train/blob/master/doc/TatoebaChallenge.md

7 Multilingual Models

One of the most interesting questions is the ability of multilingual models to push the performance of low-resource machine translation. The Tatoeba translation challenge provides a perfect testbed for systematic studies on the effect of transfer learning across various subsets of language pairs. We already started various experiments with a number of multilingual translation models that we evaluate on the given benchmarks. In our current work, we focus on models that include languages in established groups and for that we facilitate the ISO-639-5 standard. This standard defines a hierarchy of language groups and we map our data sets accordingly to start new models that cover those sets. As an example, we look at the task of Belorussian-English translation that has been included in the previous section as well. Table 2 summarizes the results of our current models sorted by chrF2 scores.

model	chr-F2	BLEU
sla-eng/opus4m	0.610	42.7
sla-eng/opus2m	0.609	42.5
sla-eng/opus1m	0.599	41.7
ine-eng/opus2m	0.597	42.2
ine-eng/opus4m	0.597	41.7
ine-eng/opus1m	0.588	41.0
zle-eng/opus4m	0.573	38.7
zle-eng/opus2m	0.569	38.3
mul-eng/opus1m	0.550	37.0
mul-eng/opus2m	0.549	36.8
zle-eng/opus1m	0.543	35.4
ine-ine/opus1m	0.512	31.8
bel-eng/opus	0.268	10.0

Table 2: Translation results of the Belorussian-English test set using various multilingual translation models compared to the baseline bilingual model (shown at the bottom). opusXm refers to sampled data sets that include X million sentences per language pair.

The models focus on different levels of relatedness of the languages and range from East Slavic Languages (zle), Slavic languages (sla) to the language family of Indo-European languages (ine) and the set that contains all languages (mul). Each model is trained on sampled data set in order to balance between different languages. The smallest training sets are based on data that are sampled to include a maximum of one million sentence per language pair (opus1m). We use both, down-sampling and up-sampling. The latter is done by simply

multiplying the existing data until the threshold is reached. We also set a threshold of 50 for the maximum of repeating the same data in order to avoid over-representing small noisy data. The one-million models are trained first and form the basis of larger models. We continue training with data sets sampled to two million before increasing to four million sentence pairs.

The Table shows some interesting patterns. First of all, we can clearly see a big push in performance when adding related languages to the training data. This is certainly expected especially in the case of Belorussian that is closely related to higher-resource-languages such as Russian and Ukrainian. Interesting is that the East Slavic language group is not the best performing model even though it includes those two related languages. The additional information from other Slavic languages pushes the performance beyond their level quite significantly. Certainly, those models will see more data and this may cause the difference. The 'sla-eng' model covers 13 source languages whereas 'zle' only 5. Also interesting to see is that the Indo-European language model fairs quite well despite the enormous language coverage that this model has to cope with. On the other hand, the big 'mul' translation model does not manage to create the same performance and the limits of the standard model with such a massive setup become apparent. Training those models becomes also extremely expensive and slow and we did not manage to start the 4-million-sentence model.

Currently, we look into the various models we train and many other interesting patterns can be seen. We will leave a careful analyses to future work and also encourage the community to explore this field further using the given collection and benchmark. Updates about models and scores will be published on the website and we would also encourage more qualitative studies that we were not able to do yet.

8 Zero-shot and few-shot translation

Finally, we have a quick look at zero-shot and few-shot translation tasks. Table 3 shows results for Awadhi-English translation, one of the test sets for which no training data is available. Awadhi is an Eastern Hindi language in the Indo-Iranian branch of the Indo-European language family.[14]

[14]We use ISO639-3 and ISO639-5 standards for names and codes of languages and language groups.

model	chr-F2	BLEU
ine-eng/opus1m	0.285	10.0
mul-eng/opus1m	0.257	9.4
inc-eng/opus1m	0.217	6.8
iir-eng/opus1m	0.214	7.9
ine-ine/opus1m	0.201	2.4
tatoeba-zero/opus	0.042	0.1

Table 3: Translation results of the Awadhi-English test set using multilingual translation models.

The table shows that a naive approach of throwing all languages that are part of zero-shot language pairs into one global multilingual model (tatoeba-zero) does not work well. This is probably not very surprising. Another interesting observation is that a symmetric multilingual model with Indo-European languages on both sides (ine-ine) also underperforms compared to other multilingual models that only translate into English. Once again, the Indo-European-language-family to English model performs quite well. Note that the performance purely comes from overlaps with related languages as no Awadhi language data is available during training. The performance is still very poor and needs to be taken with a grain of salt. They demonstrate, however, the challenges one faces with realistic cases of zero-shot translation.

In Table 4, we illustrate another case that could be described as a realistic few-shot translation task. Our collection comes with 3,613 training examples for the translation between English and Faroese. The table shows our current results in this task using multilingual models that translate from English to language groups including the Scandinavian language in question.

model	chr-F2	BLEU
eng-gem/opus	0.318	9.4
gem-gem/opus	0.312	7.0
eng-gmq/opus	0.311	7.0
eng-ine/opus	0.281	6.3
eng-mul/opus	0.280	5.7
ine-ine/opus	0.276	5.9
tatoeba-zero/opus	0.042	0.1

Table 4: Translation results of the English-Faroese test set with different multilingual NMT models.

Again, we can see that the naive tatoeba-zero model is the worst. The symmetric Indo-European model performs better but the English-Germanic

model gives the best performance, which is still very low and not satisfactory for real-world applications. Once again, the example demonstrates the challenge that is posed by extremely low-resource scenarios and we hope that the data set we provide will trigger additional fascinating studies on a large variety of interesting cases.

9 Comparison to the WMT news task

Finally, we also include a quick comparison to the WMT news translation task, see Table 5. Note that we did not perform any optimization for that task, did not use any in-domain back-translations and did not run fine-tuning in the news domain. We only give results for English–German (in both directions) for the 2019 test data to give an impression about the released baseline models.

English – German		
model	BLEU	chr-F2
eng-deu	42.4	0.664
eng-gmw	35.9	0.616
eng-gem	35.0	0.613
eng-ine	26.6	0.554
eng-mul	21.0	0.512
WMT best	44.9	–
German – English		
model	BLEU	chr-F2
deu-eng	40.5	0.645
gmw-eng	36.6	0.615
gem-eng	37.2	0.618
ine-eng	31.7	0.571
mul-eng	27.0	0.529
WMT best	42.8	–

Table 5: Translation results of baseline models on English–German news translation from WMT 2019 using bilingual and multilingual Tatoeba baseline models. The BLEU scores are also compared to the best score that is currently available from http://matrix.statmt.org/matrix – retrieved on October 4, 2020.

The results demonstrate that the models can achieve high quality even on a domain they are not optimized for. The best scores in the German–English case are close to the top performing model registered for this task even though the comparison is not fair for various reasons. The purpose is anyway not to provide state-of-the-art models for the news translation task but baseline models for the Tatoeba case and in future work we will also explore the use of our models as the basis for systems that can be developed for other benchmarks and applications. In the example we can also see that multilingual models significantly lag behind bilingual ones in high-resource cases. Each increase of the language coverage (except for the move from West Germanic languages (gmw) to Germanic languages (gem) in the German–English case) leads to a drop in performance but note that those multilingual models are not fine-tuned for translating from and to German.

10 Conclusions

This paper presents a new comprehensive data set and benchmark for machine translation that covers roughly 3,000 language pairs and over 500 languages and language variants. We provide training and test data that can be used to explore realistic low-resource scenarios and zero-shot machine translation. The data set is carefully annotated with standardized language labels including variations in scripts and with information about the original source. We also release baseline models and results and encourage the community to contribute to the data set and machine translation development. All tools for data preparation and training bilingual as well as multilingual translation models are provided as open source packages on github. We are looking forward to new models, extended test sets and a better coverage of the World's languages.

Acknowledgements

This work is supported by the FoTran project (grant agreement No 771113), funded by the European Research Council (ERC) and the MeMAD project (grant agreement No 780069) under the European Union's Horizon 2020 research and innovation program. We would also like to acknowledge the support of the CSC IT Center for Science, Finland, for computational resources.

References

Mauro Cettolo, Christian Girardi, and Marcello Federico. 2012. Wit^3: Web inventory of transcribed and translated talks. In *Proceedings of the 16th Conference of the European Association for Machine Translation (EAMT)*, pages 261–268, Trento, Italy.

Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. 2016a. Multi-way, multilingual neural machine

translation with a shared attention mechanism. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 866–875, San Diego, California. Association for Computational Linguistics.

Orhan Firat, Baskaran Sankaran, Yaser Al-onaizan, Fatos T. Yarman Vural, and Kyunghyun Cho. 2016b. Zero-resource translation with multi-lingual neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 268–277, Austin, Texas. Association for Computational Linguistics.

Thanh-Le Ha, Jan Niehues, and Alexander Waibel. 2016. Toward multilingual neural machine translation with universal encoder and decoder.

Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association of Computational Linguistics*, 5(1):339–351.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86. Citeseer.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Surafel M. Lakew, Marcello Federico, Matteo Negri, and Marco Turchi. 2018. Multilingual neural machine translation for low-resource languages. *IJCoL - Italian Journal of Computational Linguistics*, 4(1). Emerging Topics at the Fourth Italian Conference on Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Milan Straka, Jan Hajič, and Jana Straková. 2016. UDPipe: Trainable pipeline for processing CoNLL-u files performing tokenization, morphological analysis, POS tagging and parsing. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4290–4297, Portorož, Slovenia. European Language Resources Association (ELRA).

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of LREC*, Istanbul, Turkey.

Jörg Tiedemann and Santhosh Thottingal. 2020. OPUS-MT Building open translation services for the World. In *Proceedings of the 22nd Annual Conferenec of the European Association for Machine Translation (EAMT)*, Lisbon, Portugal.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

Human-Paraphrased References Improve Neural Machine Translation

Markus Freitag, George Foster, David Grangier, Colin Cherry
Google Research
{freitag,fosterg,grangier,colincherry}@google.com

Abstract

Automatic evaluation comparing candidate translations to human-generated paraphrases of reference translations has recently been proposed by Freitag et al. (2020). When used in place of original references, the paraphrased versions produce metric scores that correlate better with human judgment. This effect holds for a variety of different automatic metrics, and tends to favor natural formulations over more literal (*translationese*) ones. In this paper we compare the results of performing end-to-end system development using standard and paraphrased references. With state-of-the-art English-German NMT components, we show that tuning to paraphrased references produces a system that is significantly better according to human judgment, but 5 BLEU points worse when tested on standard references. Our work confirms the finding that paraphrased references yield metric scores that correlate better with human judgment, and demonstrates for the first time that using these scores for system development can lead to significant improvements.

1 Introduction

Machine Translation (MT) has shown impressive progress in recent years. Neural architectures (Bahdanau et al., 2015; Gehring et al., 2017; Vaswani et al., 2017) have greatly contributed to this improvement, especially for languages with abundant training data (Bojar et al., 2016, 2018; Barrault et al., 2019). This progress creates novel challenges for the evaluation of machine translation, both for human (Toral, 2020; Läubli et al., 2020) and automated evaluation protocols (Lo, 2019; Zhang et al., 2019).

Both types of evaluation play an important role in machine translation (Koehn, 2010). While human evaluations provide a gold standard evaluation, they involve a fair amount of careful and hence expensive work by human assessors. Cost therefore limits the scale of their application. On the other hand, automated evaluations are much less expensive. They typically only involve human labor when collecting human reference translations and can hence be run at scale to compare a wide range of systems or validate design decisions. The value of automatic evaluations therefore resides in their capacity to be used as a proxy for human evaluations for large scale comparisons and system development.

The recent progress in MT has raised concerns about whether automated evaluation methodologies reliably reflect human ratings in high accuracy ranges. In particular, it has been observed that the best systems according to humans might fare less well with automated metrics (Barrault et al., 2019). Most metrics such as BLEU (Papineni et al., 2002) and TER (Snover et al., 2006) measure overlap between a system output and a human reference translation. More refined ways to compute such overlap have consequently been proposed (Banerjee and Lavie, 2005; Lo, 2019; Zhang et al., 2019).

Orthogonal to the work of building improved metrics, Freitag et al. (2020) hypothesized that human references are also an important factor in the reliability of automated evaluations. In particular, they observed that standard references exhibit simple, monotonic language due to human 'translationese' effects. These standard references might favor systems which excel at reproducing these effects, independent of the underlying translation quality. They showed that better correlation between human and automated evaluations could be obtained when replacing standard references with *paraphrased* references, even when still using surface overlap metrics such as BLEU (Papineni et al., 2002). The novel references, collected by asking linguists to paraphrase standard references, were shown to steer evaluation away from rewarding

Proceedings of the 5th Conference on Machine Translation (WMT), pages 1183–1192
Online, November 19–20, 2020.

translation artifacts. This improves the assessment of alternative, but equally good translations.

Our work builds on the success of paraphrased translations for evaluating existing systems, and asks if different design choices could have been made when designing a system with such an evaluation protocol in mind. This examination has several potential benefits: it can help identify choices which improve BLEU on standard references but have limited impact on final human evaluations; or those that result in better translations for the human reader, but worse in terms of standard reference BLEU. Conversely, it might turn out that paraphrased references are not robust enough to support system development due to the presence of 'metric honeypots': settings that produce poor translations, but which are nevertheless assigned high BLEU scores.

To address these points, we revisit the major design choices of the best English→German system from WMT2019 (Ng et al., 2019) step-by-step, and measure their impact on standard reference BLEU as well as on paraphrased BLEU. This allows us to measure the extent to which steps such as data cleaning, back-translation, fine-tuning, ensemble decoding and reranking benefit standard reference BLEU more than paraphrase BLEU. Revisiting these development choices with the two metrics results in two systems with quite different behaviors. We conduct a human evaluation for adequacy and fluency to assess the overall impact of designing a system using paraphrased BLEU.

Our main findings show that optimizing for paraphrased BLEU is advantageous for human evaluation when compared to an identical system optimized for standard BLEU. The system optimized for paraphrased BLEU significantly improves WMT newstest19 adequacy ratings (4.72 vs 4.27 on a six-point scale) and fluency ratings (63.8% vs 27.2% on side-by-side preference) despite scoring 5 BLEU points lower on standard references.

2 Related Work

Collecting human paraphrases of existing references has recently been shown to be useful for system evaluation (Freitag et al., 2020). Our work considers applying the same methodology for system tuning. There is some earlier work relying on *automated* paraphrases for system tuning, especially for Statistical Machine Translation (SMT). Madnani et al. (2007) introduced an automatic paraphrasing technique based on English-to-English translation of full sentences using a statistical MT system, and showed that this permitted reliable system tuning using half as much data. Similar automatic paraphrasing has also been used to augment training data, e.g. (Marton et al., 2009), but relying on standard references for evaluation. In contrast to human paraphrases, the quality of current machine generated paraphrases degrades significantly as overlap with the input decreases (Mallinson et al., 2017; Roy and Grangier, 2019). This makes their use difficult for evaluation since (Freitag et al., 2020) suggests that substantial paraphrasing – 'paraphrase as much as possible' – is necessary for evaluation.

Our work can be seen as replacing the regular BLEU metric with a new paraphrase BLEU metric for system tuning. Different alternative automatic evaluation metric have also been considered for system tuning (He and Way, 2010; Servan and Schwenk, 2011) with Minimum Error Rate Training, MERT (Och, 2003). This work showed some specific cases where Translation Error Rate (TER) was superior to BLEU.

Our work is also related to the bias that the human translation process introduces in the references, including source language artifacts—*Translationese* (Koppel and Ordan, 2011)—as well as source-independent artifacts—*Translation Universals* (Mauranen and Kujamäki, 2004). The professional translation community studies both systematic biases inherent to translated texts (Baker, 1993; Selinker, 1972), as well as biases resulting specifically from interference from the source text (Toury, 1995). For MT, Freitag et al. (2019) point at Translationese as a source of mismatch between BLEU and human evaluation, raising concerns that overlap-based metrics might reward hypotheses with translationese language more than hypotheses using more natural language. The impact of Translationese on human evaluation of MT has recently received attention as well (Toral et al., 2018; Zhang and Toral, 2019; Graham et al., 2019). More generally, the question of bias to a specific reference has also been raised, in the case of monolingual manual evaluation (Fomicheva and Specia, 2016; Ma et al., 2017). Different from the impact of Translationese on evaluation, the impact of Translationese in the training data has also been studied (Kurokawa et al., 2009; Lembersky et al., 2012a; Bogoychev and Sennrich, 2019; Riley et al., 2020).

Finally, our work is also related to studies measuring the importance of the test data quality, looking specifically at the test set translation direction. For SMT evaluation, Lembersky et al. (2012b) and Stymne (2017) explored how the translation direction affects translation results. Holmqvist et al. (2009) noted that the original language of the test sentences influences the BLEU score of translations. They showed that the BLEU scores for target-original sentences are on average higher than sentences that have their original source in a different language. Recently, a similar study was conducted for neural MT (Bogoychev and Sennrich, 2019).

3 Experimental Setup

We first describe data and models, then present our human evaluation protocol.

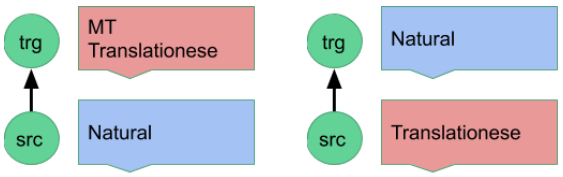
3.1 Data

We ran all experiments on the WMT 2019 English→German news translation task (Barrault et al., 2019). The task provides ∼38M parallel sentences. As German monolingual data, we concatenate all News Crawl data from 2007 to 2018, comprising ∼264M sentences after removing duplicates.

In addition to the training data, we use newstest2018 for development and newstest2019 for evaluation only. There is an important difference between these two test sets. Newstest2018 was created from monolingual news data from both English and German online sources. Half of the data consists of English text translated into German, while the other half consists of German text translated into English. This results in a joint test set of 2,998 sentences. Newstest2019, on the other hand, consists only of 1,997 sentences translated from English into German (see Figure 1). To provide a joint test set similar to newstest2018, we took newstest2019 from the reverse translation direction German→English, swapped source and target, and concatenated it with the original test sets. This results in a new joint newstest2019 test set of 3,997 sentences.

In addition to reporting overall BLEU scores on the different test sets, we also report results on the two subsets (based on the original language) of each newstest20XX, which we call the *orig-en* and the *orig-de* halves of the test set.

Freitag et al. (2020) provided an alternative reference translation for the orig-en half of newstest2019. For both standard and alternative references, they provided an additional paraphrased 'as much as possible' version (four different references in all). In order to enable our parameter tuning experiments, we created a paraphrased version of the reference for the orig-en half of *newstest2018* (1,500 sentences) following the instructions from Freitag et al. (2020). We will release this new paraphrased reference, *newstest2018.orig-en.p*, as part of our work.



(a) Forward-translated, i.e. source original

(b) Backward-translated, i.e. target original

Figure 1: Sentences in a test set are either natural in the source and forward-translated into the target language, or vice-versa. If a test set consists of both kinds of sentences, we call it a joint test set. WMT English→German newstest2018 is a joint test set with half of the sentences being forward-translated. WMT English→German newstest2019 is a forward-translated test set.

3.2 Models

For our translation models, we adopt the transformer implementation from *Lingvo* (Shen et al., 2019), using the transformer-big model size (Vaswani et al., 2017). We use a vocabulary of 32k subword units and exponentially moving averaging of checkpoints (EMA decay) with the weight decrease parameter set to $\alpha = 0.999$ (Buduma and Locascio, 2017). We used a batch size of around 32k sentences in all our experiments.

We report BLEU (Papineni et al., 2002) in addition to human evaluation. All BLEU scores are calculated with sacreBLEU (Post, 2018)[1].

3.3 Human Evaluation

To collect human rankings, we ran side-by-side evaluation for overall quality and fluency. We hired 20 linguists and divided them equally between the two evaluations. Each evaluation included 1,000 items with each item being rated exactly once. We acquired only a single rating per sentence from the professional linguists as we found that they were

[1]BLEU+case.mixed+lang.ende+numrefs.1+smooth.exp+SET+tok.13a+version.1.4.12 SET ∈{wmt18, wmt19, wmt19/google/ar, wmt19/google/arp, wmt19/google/wmtp}

more reliable than crowd workers (Toral, 2020). We evaluated the orig-en sentences corresponding to the official WMT-19 English→German test set (Barrault et al., 2019). Results in this natural translation direction are more meaningful as pointed out by Zhang and Toral (2019), who show that translating a 'translationese' source is simpler and should not be used for human evaluation.
Our human evaluation followed the protocol:

- Fluency: We present two translations of the same source sentence to professional linguists without showing the actual source sentence. We then ask the rater wether they prefer one of the outputs or rate them equally based on fluency.
- Overall Quality: We present two translations along with the source and ask the raters to evaluate each translation on a 6-point scale. A score of 6 will be assigned to translations with 'perfect meaning and grammar', while a score of 0 will be assigned to 'nonsense/ no meaning preserved' translations. The average over all ratings yields the system's final quality score.

4 Experimental Results

This section first presents our main result comparing the same system tuned with BLEU on standard versus paraphrased references. We then break down how system design choices impact each metric differently. Throughout, we refer to scores computed with standard references as BLEU, and those computed with paraphrased references as BLEUP.

4.1 Overall Performance

We compare the performance of a system optimized on newstest2018 with standard references (opt-on-BLEU) with one optimized on newstest2018.orig-en with paraphrased references (opt-on-BLEUP). Both systems were developed using only newstest2018 data, keeping newstest2019 as a blind test set. Table 1 summarizes the results on newstest2019. Details of how these two systems were developed and how they differ are given in Section 4.2.

The opt-on-BLEU system outperforms opt-on-BLEUP by 5.2 BLEU points. Normally this would lead us to discard opt-on-BLEUP. However, the BLEUP scores tell a different story: opt-on-BLEUP outperforms by 0.3 points, a potentially large improvement given the smaller natural range of this metric. Under a significance test with random approximation (Riezler and Maxwell III, 2005), both the BLEU and BLEUP differences are significant at p<5e-18.

	opt-on-BLEU	opt-on-BLEUP
BLEU	**45.0**	39.8
BLEUP	13.4	**13.7**
human quality	4.27	**4.72**
human fluency	27.2%	**63.8%**

Table 1: BLEU scores and human ratings for WMT newstest2019 English→German (original English sources). We optimized the system to perform best on either newstest2018 with standard reference translations (opt-on-BLEU) or newstest2018.orig-en with paraphrased reference translations (opt-on-BLEUP). BLEU differences are significant according to random approximation (Riezler and Maxwell III, 2005) with p<5e-18. Human score differences are significant according to a Wilcoxon rank-sum test with p<5e-18.

Freitag et al. (2020) showed that BLEU scores calculated on paraphrased references have higher correlation with human judgment than those calculated on standard references. To verify their findings, we ran a human evaluation for the two different outputs on 1,000 sentences randomly drawn from newstest2019 (orig-en), as described above. As shown in Table 1, opt-on-BLEUP is consistently evaluated as better for both quality and fluency. To measure the significance between the two ratings, we ran a Wilcoxon rank sum test on the human ratings and found that both improvements are significant with p<e-18.

This experiment demonstrates that we can actually tune our MT system on paraphrased references to yield higher translation quality when compared to a typical system tuned on standard BLEU. Interestingly, the BLEU score for the better system is much lower, supporting our contention that BLEU rewards spurious translation features (e.g. monotonicity and common translations) that are filtered out by BLEUP.

4.2 Analysing Performance

We now describe the individual model decisions that went into the two final systems of Section 4.1. To build a classical system optimized on BLEU with standard references, we replicate the WMT 2019 winning submission (Ng et al., 2019) and examine

the effect of each of its major design decisions.[2] In particular, we are looking into the effect of data cleaning, back-translation, fine tuning, ensembling and noisy channel reranking. We examine the impact of each method on BLEU and BLEUP. For our experiments, we used newstest2018 as our development set and newstest2019 as our held-out test set. All model decisions (checkpoint, variants) are solely made on newstest2018.

Experimental results are presented in Table 2. As described in Section 3.1, we report 4 different BLEU scores for newstest2018 (dev) and newstest2019 (test). In addition to reporting BLEU score on the joint or the orig-de/orig-en halves of the test sets, we also report BLEU scores that are calculated on paraphrased references (BLEUP).

4.2.1 Data Cleaning

For data cleaning, we used CDS (Wang et al., 2018). We trained a CDS model for English→German taking news-commentary as the in-domain/clean data set. We scored all parallel sentences with our trained CDS model and kept the 70% highest scoring sentences. Our experimental results suggest that data cleaning is useful for all four types of test sets and consistently improves over a baseline system that is trained on raw parallel data. We conclude that data cleaning is useful for all systems independently of which test set it will be optimized for.

4.2.2 Back-Translation

We trained a strong German→English model on the same parallel data (with flipped source/target) and used that model to (back-)translate (BT) all deduped German monolingual sentences from NewsCrawl 2007-2018 into English. We filtered sentences with a source-target ratio lower than 0.5 or higher than 1.5. We further run language identification and filtered out all backtranslations going into the wrong language. We then oversample our bitext data to match the size of the backtranslation data and train a NMT model on the concatenation of both datasets.

As previously reported by (Freitag et al., 2019; Bogoychev and Sennrich, 2019), the original language of the sentences within a test is crucial and can lead to very different conclusions, in particular for back-translation systems. This difference is visible when looking at the BLEU scores on the standard references. While the BLEU score on orig-de does improve by 7.5 points, the BLEU score drops by 2.9 points on the orig-en half. Due to the big gain on the orig-de half, BT also improves the BLEU score on the joint set. The paraphrased references were designed to overcome these kinds of mismatches and they show a gain of 0.5 BLEU points. We can conclude that back-translation helps improve BLEU and BLEUP and we include BT for systems that are optimized for both standard or paraphrased BLEU scores.

4.2.3 Fine-Tuning

Similar to (Ng et al., 2019), we fine-tuned our back-translated model on a concatenation of previous WMT testsets (newstest{2013,2015,2016,2017}) and the clean in-domain news-commentary corpus. In total, we fine-tuned the model on 330k sentences. We kept all model parameters the same (batch size, learning rate) and continued training on the fine-tuned data for one epoch. The BLEU scores on the standard references suggest a small improvement of 0.3 BLEU on the joint test set. Interestingly, the improvement is visible on the orig-en half by 0.7 points while the BLEU scores on orig-de actually drop by 1.7 points. Nevertheless, BLEUP does improve by 0.5 points, suggesting that fine-tuning is especially helpful when measuring scores with paraphrased references. Despite the small gain on standard references, we include fine-tuning in both our optimized systems.

4.2.4 Ensemble

Combining different predictions is a standard approach in MT to boost BLEU scores. We run ensemble decoding with 4 previously built models. In addition to using the 3 models described in Section 4.2.1, 4.2.2, and 4.2.3, we build a second fine-tuned model with the same approach, but different initialization.

Although ensemble decoding improves the performance on our standard references by up to 1.9 BLEU points, the quality is rated as lower by 0.3 BLEU points on the paraphrased references. We suspect that using an ensemble for decoding favors common, average language by promoting target spans where all systems agree. Paraphrase translations actually downweight the importance of this language, which seems important for agreeing with human judgments (Freitag et al., 2020). This promotion of average language and monotonic translation may explain the effectiveness of ensembling

[2] Our replication achieves 45.0 BLEU on newstest19, competitive with the reference system at 42.7 BLEU.

	newstest2018 (dev)				newstest2019 (test)			
	joint	orig-de	orig-en	orig-en.p	joint	orig-de	orig-en	orig-en.p
(1) bitext	46.0	38.8	50.6	12.8	38.5	34.9	40.9	12.1
(2) + CDS	46.1	39.4	50.5	13.4	39.6	35.6	42.3	12.6
(3) + BT	47.2	45.3	47.7	13.6	40.9	43.1	39.4	13.1
(4) + Fine tuning	47.7	43.6	49.2	13.8	41.2	41.3	41.1	13.6
(5) + Ensemble of 4	49.8	45.4	52.1	13.7	43.1	42.1	43.6	13.3
+ reranking of (5) (opt on BLEU)	**50.7**	44.8	53.9	13.8	**43.4**	41.2	45.0	13.4
+ reranking of (4) (opt on BLEUP)	47.1	45.9	47.1	**14.7**	41.6	44.0	39.8	**13.7**

Table 2: BLEU scores for WMT 2019 English→German. The *joint* sets combine *orig-en* and *orig-de* subsets. The *orig-en.p* sets use paraphrased references instead of standard references. Our experiments compared *newstest2018.joint* and *newstest2018.orig-en.p* for system tuning. The standard newstest2018 and newstest2019 sets are *newstest2018.joint* and *newstest2019.orig-en*, respectively.

only for standard reference BLEU. Similar to the WMT 2019 winning submission, we include the ensemble approach in our system that is optimized on the joint BLEU scores. However, we do not include it in our system optimized on BLEUP.

4.3 Reranking

Finally, we extend the noisy-channel approach (Yee et al., 2019) which consists of re-ranking the top-50 beam search output of either the ensemble model (when tuned for BLEU) or the fine-tuned model (when tuned for BLEUP). Instead of using 4 features—forward probability, backward probability, language model and word penalty—we use 11 forward probabilities, 10 backward probabilities and 2 language model scores. Different to (Ng et al., 2019), we did not pick the re-ranking weights through random search, but used MERT (Och, 2003) for efficient tuning.

The 11 different forward translation scores come from different English→German NMT models that are replicas of the previous described models (Section 4.2.1, 4.2.2, and 4.2.3). The 10 backward translation scores come from the same approaches, but trained in the reverse direction. These 21 NMT model scores are combined with 2 language model (LM) scores. The first LM is trained on the German monolingual NewsCrawl data, while the second LM is trained on forward-translated English NewsCrawl data. The first LM should assign high scores to genuine German text, while the second LM should assign high scores to translationese German originating from English.

We first reranked the 50-best list generated by the ensemble model with MERT on newstest2018. Similar to the original WMT 2019 submission, the BLEU scores on the joint and orig-en set increase. This reranked output corresponds to our opt-on-BLEU model. Next, we reranked the 50-best list generated by the fine-tuned model with MERT on newstest2018.orig-en with paraphrased references. This led to further small increases in BLEUP, and corresponds to our opt-on-BLEUP model.

In summary, optimizing on BLEUP leads us to keep back-translation, even though evaluation with standard English-original references would have us drop it, and also leads us to drop the ensembling step. Rescoring using MERT weights learned with BLEU or BLEUP further separates the systems according to these metrics.

5 Analysis

This section confirms the results from the previous section with additional references for newstest2019 and illustrates the behaviour of our systems on individual sentences.

5.1 Alternative Reference Translations

Freitag et al. (2020) released an additional standard reference translation (AR) and two ‘paraphrase as-much-as-possible‘ reference translations for newstest2019 (WMT.p and AR.p). We used WMT.p in all our above experiments; here we report BLEU scores for all four available reference translations in table 3. The BLEU improvements between the two standard reference translations agree perfectly. Similarly, the BLEUP improvements between the two paraphrased references also coincide. This indicates that by optimizing on BLEU or BLEUP we have not somehow overfit to a specific set of reference translations or their paraphrases, but instead

have molded our model to better match a style of reference translation.

5.2 Translation Examples

This section presents translation examples from our two differently optimized systems in Table 4. The first 3 examples show sentences where opt-on-BLEUP has higher translation quality than opt-on-BLEU. One observation of (Freitag et al., 2020) was that BLEU scores calculated on standard references prefer monotonic translations. This is visible in our first translation example, where opt-on-BLEU incorrectly translates the saying *Tomorrow's a different beast* into *Morgen ist ein anderes Biest*, using an inappropriately monotonic strategy. On the other hand, the opt-on-BLEUP system captures the meaning of the source sentence and generates a valid translation.

Another drawback of standard reference BLEU is the preference for literal translation. This is visible in our second example where the word *cap* is translated into *Kappe* and *tip* into *kippen*. Both are valid word-by-word translations, but do not make much sense in this context. The third example is another example of the monotonic translation style of a regular tuned system. The opt-on-BLEU translation is an incorrect word-by-word translation. The opt-on-BLEUP system is able to introduce a German natural sentence structure and generate a flawless translation.

The last translation example is a loss for the paraphrased-tuned system and demonstrates that sometimes a more literal translation can be better. Even though the word *run* can be translated into *Ansturm*, it is not appropriate in this context and the simpler translation *Lauf* is correct.

5.3 Matched n-grams

The BLEU scores calculated on the two different references yield different conclusions. BLEU on standard references evaluated opt-on-BLEU higher by more than 5 BLEU points. BLEUP came to a different conclusion and gave a higher score to opt-on-BLEUP. In this section, we look at the n-grams that contributed most to these different outcomes. Those that contribute most to the difference in BLEU across the two systems are:

- **Er sagte, dass** (*He said that*)
- **, sagte er der** (*, he said the*)
- **stellte fest, dass** (*noted that*)

These are all generic, high-frequency n-grams. They are crucial for attaining high BLEU scores, and tend to appear in translations that employ the same structure as the source sentence. In contrast, the n-grams that contribute most to the difference in BLEUP are:

- **Menschen ums Leben kamen** (*humans died*)
- **Grossbritanien keine Steuern zahlen** (*Great Britain pay no tax*)
- **von BBC Scottland** (*from BBC Scottland*)

These are much less frequent sequences with more semantic content.

6 Conclusions

Prior work has shown that BLEU measured on paraphrased references (BLEUP) has better correlation with human evaluation than BLEU measured on regular references (BLEU) for the comparison of *existing* systems (Freitag et al., 2019). Motivated by this finding, we collected a development set of paraphrased references and assessed BLEUP for *system development*. This allowed us to evaluate if the design choices of a modern neural MT system impact BLEU and BLEUP differently, including tuning a re-ranking noisy channel model to these metrics. Our experiments followed the setup from the winning newstest19 English→Germam entry at WMT19 (Ng et al., 2019).

For design choices, we observe that BLEUP seems to emphasize the importance of back-translation even when test sets are source original. On the other end, BLEUP seems to de-emphasize the importance of ensembles, as the reliable prediction of common language by ensembles is less rewarded by this metric.

Our tuning experiments led to positive results. In human evaluation, the system tuned on BLEUP showed significant improvements in terms of adequacy and even greater gains in terms of fluency compared to the system tuned on BLEU. Example translations indicate that the model tuned on BLEUP produces noticeably less literal translations. Our experiments also highlight a disconnect between regular BLEU and human evaluation: the system tuned on BLEUP degrades standard BLEU scores by over 5 points, while faring significantly better in human evaluation. Paraphrased automatic evaluation therefore seems to be a promising proxy

	newstest2019			
	WMT (orig-en)	AR (orig-en)	WMT.p (orig-en.p)	AR.p (orig-en.p)
(1) bitext	40.9	32.2	12.1	12.0
(2) + CDS	42.3	34.2	12.6	12.3
(3) + BT	39.4	33.6	13.1	13.0
(4) + Fine tuning	41.1	35.5	13.6	13.4
(5) + Ensemble of 4	43.6	36.0	13.3	13.0
+ reranking of (5) (opt-on-BLEU)	**45.0**	**36.7**	13.4	13.1
+ reranking of (4) (opt-on-BLEUP)	39.8	34.4	**13.7**	**13.5**

Table 3: BLEU scores for English→German newstest2019 for the additional references from (Freitag et al., 2020).

source	Tomorrow's a different beast.
opt on BLEU	Morgen ist ein anderes Biest.
opt on BLEUP	Morgen ist alles anders.
source	You have to tip your cap.
opt on BLEU	Sie müssen Ihre Kappe kippen.
opt on BLEUP	Man muss den Hut ziehen.
source	He averaged 5.6 points and 2.6 rebounds a game last season.
opt on BLEU	Er durchschnittlich 5,6 Punkte und 2,6 Rebounds ein Spiel in der vergangenen Saison.
opt on BLEUP	In der vergangenen Saison erzielte er im Schnitt 5,6 Punkte und 2,6 Rebounds pro Spiel.
source	Thirty-two percent supported such a run.
opt on BLEU	32 Prozent unterstützten einen solchen Lauf.
opt on BLEUP	32 Prozent sprachen sich für einen solchen Ansturm aus.

Table 4: Example output for English→German for systems optimized on standard BLEU or BLEUP. Translations for opt-on-BLEU tend to be more literal, and adhere closely to the source sentence structure.

for human evaluation when making design choices for MT systems.

This research opens the question of whether these results can be confirmed over a wide range of language pairs. We also hope to achieve further improvements by refining the paraphrased evaluation protocol.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Mona Baker. 1993. Corpus Linguistics and Translation Studies: Implications and Applications. *Text and technology: in honour of John Sinclair*, pages 233–252.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In *Proceedings of the ACL workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

Loïc Barrault, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. 2019. Findings of the 2019 Conference on Machine Translation (WMT19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 1–61, Florence, Italy. Association for Computational Linguistics.

Nikolay Bogoychev and Rico Sennrich. 2019. Domain, Translationese and Noise in Synthetic Data for Neural Machine Translation. *arXiv preprint arXiv:1911.03362*.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurélie Névéol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. 2016. Findings of the 2016 Conference

on Machine Translation. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 131–198, Berlin, Germany. Association for Computational Linguistics.

Ondřej Bojar, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Philipp Koehn, and Christof Monz. 2018. Findings of the 2018 Conference on Machine Translation (WMT18). In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 272–303, Belgium, Brussels. Association for Computational Linguistics.

Nikhil Buduma and Nicholas Locascio. 2017. *Fundamentals of deep learning: Designing next-generation machine intelligence algorithms*. "O'Reilly Media, Inc.".

Marina Fomicheva and Lucia Specia. 2016. Reference bias in monolingual machine translation evaluation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 77–82, Berlin, Germany. Association for Computational Linguistics.

Markus Freitag, Isaac Caswell, and Scott Roy. 2019. APE at Scale and Its Implications on MT Evaluation Biases. In *Proceedings of the Fourth Conference on Machine Translation*, pages 34–44, Florence, Italy. Association for Computational Linguistics.

Markus Freitag, David Grangier, and Isaac Caswell. 2020. BLEU might be Guilty but References are not Innocent. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, Online. Association for Computational Linguistics.

Jonas Gehring, Michael Auli, David Grangier, and Yann Dauphin. 2017. A convolutional encoder model for neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 123–135, Vancouver, Canada. Association for Computational Linguistics.

Yvette Graham, Barry Haddow, and Philipp Koehn. 2019. Translationese in machine translation evaluation.

Yifan He and Andy Way. 2010. Metric and reference factors in minimum error rate training. *Machine Translation*, 24(1):27–38.

Maria Holmqvist, Sara Stymne, Jody Foo, and Lars Ahrenberg. 2009. Improving Alignment for SMT by Reordering and Augmenting the Training Corpus. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 120–124. Association for Computational Linguistics.

Philipp Koehn. 2010. *Statistical Machine Translation*. Cambridge University Press.

Moshe Koppel and Noam Ordan. 2011. Translationese and Its Dialects. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, pages 1318–1326.

David Kurokawa, Cyril Goutte, and Pierre Isabelle. 2009. Automatic detection of translated text and its impact on machine translation. In *Proceedings of MT-Summit XII*, pages 81–88.

Samuel Läubli, Sheila Castilho, Graham Neubig, Rico Sennrich, Qinlan Shen, and Antonio Toral. 2020. A set of recommendations for assessing human–machine parity in language translation. *Journal of Artificial Intelligence Research*, 67:653–672.

Gennadi Lembersky, Noam Ordan, and Shuly Wintner. 2012a. Adapting Translation Models to Translationese Improves SMT. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '12, pages 255–265, Stroudsburg, PA, USA. Association for Computational Linguistics.

Gennadi Lembersky, Noam Ordan, and Shuly Wintner. 2012b. Language models for machine translation: Original vs. translated texts. *Computational Linguistics*, 38(4):799–825.

Chi-kiu Lo. 2019. Yisi-a unified semantic mt quality evaluation and estimation metric for languages with different levels of available resources. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 507–513.

Qingsong Ma, Yvette Graham, Timothy Baldwin, and Qun Liu. 2017. Further investigation into reference bias in monolingual evaluation of machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2476–2485, Copenhagen, Denmark. Association for Computational Linguistics.

Nitin Madnani, Necip Fazil Ayan, Philip Resnik, and Bonnie J Dorr. 2007. Using paraphrases for parameter tuning in statistical machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 120–127. Association for Computational Linguistics.

Jonathan Mallinson, Rico Sennrich, and Mirella Lapata. 2017. Paraphrasing revisited with neural machine translation. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, Valencia, Spain.

Yuval Marton, Chris Callison-Burch, and Philip Resnik. 2009. Improved statistical machine translation using monolingually-derived paraphrases. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1*, EMNLP '09, pages 381–390, Stroudsburg, PA, USA. Association for Computational Linguistics.

Anna Mauranen and Pekka Kujamäki. 2004. *Translation universals: Do they exist?*, volume 48. John Benjamins Publishing.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook fair^{TM}s wmt19 news translation task submission. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 314–319, Florence, Italy. Association for Computational Linguistics.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st annual meeting of the Association for Computational Linguistics*, pages 160–167.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Matt Post. 2018. A Call for Clarity in Reporting Bleu Scores. *arXiv preprint arXiv:1804.08771*.

Stefan Riezler and John T Maxwell III. 2005. On some pitfalls in automatic evaluation and significance testing for mt. In *Proceedings of the ACL workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 57–64.

Parker Riley, Isaac Caswell, Markus Freitag, and David Grangier. 2020. Translationese as a language in "multilingual" NMT. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7737–7746, Online. Association for Computational Linguistics.

Aurko Roy and David Grangier. 2019. Unsupervised Paraphrasing without Translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6033–6039. Association for Computational Linguistics.

Larry Selinker. 1972. Interlanguage. *International Review of Applied Linguistics*, pages 209–241.

Christophe Servan and Holger Schwenk. 2011. Optimising multiple metrics with mert. *The Prague Bulletin of Mathematical Linguistics*, 96(1):109–117.

Jonathan Shen, Patrick Nguyen, Yonghui Wu, Zhifeng Chen, Mia X. Chen, Ye Jia, Anjuli Kannan, Tara N. Sainath, and Yuan Cao et al. 2019. Lingvo: a Modular and Scalable Framework for Sequence-to-Sequence Modeling. *CoRR*, abs/1902.08295.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of association for machine translation in the Americas*. Cambridge, MA.

Sara Stymne. 2017. The Effect of Translationese on Tuning for Statistical Machine Translation. In *The 21st Nordic Conference on Computational Linguistics*, pages 241–246.

Antonio Toral. 2020. Reassessing Claims of Human Parity and Super-Human Performance in Machine Translation at WMT 2019. *arXiv preprint arXiv:2005.05738*.

Antonio Toral, Sheila Castilho, Ke Hu, and Andy Way. 2018. Attaining the Unattainable? Reassessing Claims of Human Parity in Neural Machine Translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 113–123, Belgium, Brussels. Association for Computational Linguistics.

Gideon Toury. 1995. *Descriptive Translation Studies and Beyond*. Benjamins translation library. John Benjamins Publishing Company.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Wei Wang, Taro Watanabe, Macduff Hughes, Tetsuji Nakagawa, and Ciprian Chelba. 2018. Denoising neural machine translation training with trusted data and online data selection. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 133–143, Belgium, Brussels. Association for Computational Linguistics.

Kyra Yee, Nathan Ng, Yann N Dauphin, and Michael Auli. 2019. Simple and effective noisy channel modeling for neural machine translation. *arXiv preprint arXiv:1908.05731*.

Mike Zhang and Antonio Toral. 2019. The effect of translationese in machine translation test sets. *CoRR*, abs/1906.08069.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with BERT. *Arxiv*, 1904.09675.

Incorporating Terminology Constraints in Automatic Post-Editing

David Wan[1], **Chris Kedzie**[1], **Faisal Ladhak**[1], **Marine Carpuat**[2] **and Kathleen McKeown**[1]
[1] Columbia University, [2] University of Maryland
dw2735@columbia.edu, {kedzie,faisal,kathy}@cs.columbia.edu, marine@cs.umd.edu

Abstract

Users of machine translation (MT) may want to ensure the use of specific lexical terminologies. While there exist techniques for incorporating terminology constraints during inference for MT, current APE approaches cannot ensure that they will appear in the final translation. In this paper, we present both autoregressive and non-autoregressive models for lexically constrained APE, demonstrating that our approach enables preservation of 95% of the terminologies and also improves translation quality on English-German benchmarks. Even when applied to lexically constrained MT output, our approach is able to improve preservation of the terminologies. However, we show that our models do not learn to copy constraints systematically and suggest a simple data augmentation technique that leads to improved performance and robustness.

1 Introduction

Automatic post-editing (APE) aims to improve the quality of the output of an arbitrary machine translation (MT) system by pruning systematic errors and adapting to a domain-specific style and vocabulary (Simard et al., 2007; Chatterjee et al., 2018). Although previous work has shown the usefulness of APE to prune errors by focusing on improving the translation error rate (TER), few have studied the effect of incorporating lexical constraints.

There are several use cases where such a system would be beneficial. For example, content providers meticulously curate lists of terminologies for their domains that indicate preferred translations for technical terms. Lexically constrained APE would also be useful for cross-lingual information retrieval. When displaying snippets from retrieved documents, the query term should appear in the translation output (if it does in the source) as it can make relevance clear to the end user. Here, the query serves as the term.

While recent approaches allow inference time adaptation of NMT systems using these terminologies (Dinu et al., 2019; Post and Vilar, 2018), post-editing translations with a generic APE system may lead to dropped terms. A constraint-aware APE system would allow to fix systematic translation errors, while keeping the terminologies intact.

Inspired by Dinu et al. (2019), we consider a range of representations which augment input sequences with constraint tokens and factors for use in an autoregressive Transformer (AT) APE model. Using this approach, the constraints are explicitly represented in the encoder input sequence, and the model learns to prefer translations that contain the supplied terminologies during decoding. We also explore the use of the Levenshtein Transformer (LevT) (Gu et al., 2019), a non-autoregressive Transformer (NAT) model. The LevT model applies neatly to the APE task since the decoder can be initialized with an incomplete sequence to be refined. Additionally, multiple corrections can be made simultaneously, yielding a decoding speedup over autoregressive models.

We then show that constrained APE improves translation quality and terminology preservation on top of both unconstrained and constrained MT. While both constrained and unconstrained APE models perform similarly on reducing systemic errors in the MT output, they differ in their ability to preserve terminology constraints. When applying unconstrained APE on top of constrained MT, we find a 12.6% relative drop of supplied terminology constraints as compared to a fully constrained MT to APE pipeline.

We experiment extensively with variations of both AT and LevT models, testing on both PBMT and NMT English to German WMT APE tasks (Chatterjee et al., 2018). Under all scenarios, the model performs post-editing while satisfying terminology constraints when supplied.

Proceedings of the 5th Conference on Machine Translation (WMT), pages 1193–1204
Online, November 19–20, 2020.

Our evaluation of both constrained AT and NAT models on PBMT and NMT APE benchmarks shows that both models correctly translate more than 95% of terminology constraints, with the NAT model achieving the highest coverage of terminologies at the expense of post-editing quality.

Finally, using constraints constructed with synonyms and antonyms, we show that our models do not learn to copy constraints systematically, and introduce a simple data augmentation strategy to improve the preservation of unusual constraints.

To summarize, our contributions are as follows:

1. We propose the terminology constrained APE task and evaluate several AT and LevT model variants for incorporating lexical constraints.

2. We empirically show that constrained APE is necessary to preserve terminology constraints in a MT to APE pipeline.

3. We analyze the robustness of the constraint translation behavior and suggest a simple data augmentation technique that both improves translation quality and increases the number of correctly translated terms.

2 Related Work

2.1 MT with Terminology Constraints

Integrating terminology constraints into translation can be divided into two approaches: constrained decoding and input sequence modification.

Constrained decoding modifies the decoding process to enforce the generation of the specified terminologies. This includes methods that modify beam search, such as grid beam search (Hokamp and Liu, 2017) and dynamic beam allocations (Post and Vilar, 2018). While these approaches are effective in including terminologies, they come with an increase in inference time due to the added overhead in the search algorithm.

The LevT (Gu et al., 2019), which uses a non-autoregressive decoding procedure, can initialize its decoder with a partial or incomplete output sequence. By initializing the decoder output with terminology constraints, Susanto et al. (2020) train a LevT model to perform constrained decoding. Unlike constrained search methods in autoregressive models, this initialization technique does not add any significant overhead to the decoding process. When modified to disallow deletion of terms and insertion between consecutive terminology tokens, LevT is able to retain all terminologies without affecting the performance and speed.

Alternatively, Dinu et al. (2019) propose modifying the encoder input sequence to represent terminology constraints. During training, the model learns to identify constraints in the input sequence, and translate them appropriately during decoding. This approach has the benefit of not adding additional overhead during inference.

2.2 Automatic Post-Editing

The APE task has gone through many iterations, since it was originally proposed by Simard et al. (2007). Initially, the task was to improve an unknown phrase-based machine transition (PBMT) system. An additional task to fix errors of an NMT system was introduced at WMT 2018 (Chatterjee et al., 2018).

For the APE tasks, the use of the multi-source variant of the neural encoder-decoder model is the most popular approach (Bojar et al., 2017), with the Multi-source Transformer (MST) instantiation (Junczys-Dowmunt and Grundkiewicz, 2018) achieving state-of-the-art results in 2018. Based on the AT model (Vaswani et al., 2017), the MST model consists of two Transformer encoders and a single decoder. The source sentence and the MT system output are fed separately to the two encoders, where the outputs are concatenated and then fed into the decoder to perform post-editing.

Recent work has explored alternative architectures for APE. The winner of 2019 APE tasks (Lopes et al., 2019), for example, uses a BERT-based encoder and decoder. Gu et al. (2019) both introduce the LevT model and demonstrate its utility on an APE task.

3 Constrained APE

The task of APE is to correct systematic errors in an MT system output. An APE model takes as input two sequences: the source language sentence to be translated and the translation of this sentence into the target language by an MT system. The intended output is a corrected version of the MT system's initial translation (Simard et al., 2007).

Constrained APE allows for the specification of terminology constraints: a translation for one or more phrases in the source language input may be pre-specified as additional input. The constrained APE model must use the supplied terminology constraints when performing the APE task.

Source ($\mathbf{x}$)	The Gradient tool also provides most of the same **features** as the Gradient panel .
Append ($\mathbf{x}^+$)	The$_0$ Gradient$_0$ tool$_0$ also$_0$ provides$_0$ most$_0$ of$_0$ the$_0$ same$_0$ **features**$_1$ **Funktionen**$_2$ as$_0$ the$_0$ Gradient$_0$ panel$_0$.$_0$
Replace ($\mathbf{x}^-$)	The$_0$ Gradient$_0$ tool$_0$ also$_0$ provides$_0$ most$_0$ of$_0$ the$_0$ same$_0$ **Funktionen**$_2$ as$_0$ the$_0$ Gradient$_0$ panel$_0$.$_0$
MT ($\boldsymbol{\gamma}$)	Das$_3$ Verlaufswerkzeug$_3$ bietet$_3$ außerdem$_3$ die$_3$ meisten$_3$ der$_3$ gleichen$_3$ Merkmale$_3$ wie$_3$ das$_3$ Verlaufsbedienfeld$_3$.$_3$
Post-Edit ($\mathbf{y}$)	Das Verlaufswerkzeug bietet fast dieselben **Funktionen** wie das Verlaufsbedienfeld .

Figure 1: An example of the inputs and output for the constrained APE task. Source is the source sentence. Post-Edit is the corrected MT sentence. We show the *Append* and *Replace* method to incorporate terminologies on the source side. Factors indicated for each word as source word (0), source constraint (1), target constraint (2), and MT word (3). The terminology pair for this example is (*features*, *Funktionen*).

Formally, let $\mathbf{x} = [x_1, \ldots, x_m]$ and $\boldsymbol{\gamma} = [\gamma_1, \ldots, \gamma_n]$ be the source language sentence and the initial MT translation respectively. The tokens x_i and γ_i are drawn from the source and target language vocabularies $\mathcal{S}$ and $\mathcal{T}$ respectively. The target post-edited sentence is a sequence $\mathbf{y} = [y_1, \ldots, y_o]$, with tokens y_i also drawn from $\mathcal{T}$.

We are also given a series of t translation constraints, $\mathcal{C} = \left\{\left(\check{\boldsymbol{x}}^{(1)}, \check{\boldsymbol{y}}^{(1)},\right), \ldots, \left(\check{\boldsymbol{x}}^{(t)}, \check{\boldsymbol{y}}^{(t)},\right)\right\}$, where each constraint $\left(\check{\boldsymbol{x}}^{(i)}, \check{\boldsymbol{y}}^{(i)}\right) \in \mathcal{S}^* \times \mathcal{T}^*$ is a tuple of source language phrase, $\check{\boldsymbol{x}}^{(i)} = [\check{x}_1, \ldots, \check{x}_j]$, and its desired translation, $\check{\boldsymbol{y}}^{(i)} = [\check{y}_1, \ldots, \check{y}_k]$, into the target language.

The goal of the constrained APE task is to learn a mapping of $\mathbf{x}, \boldsymbol{\gamma}$, and $\mathcal{C}$ to the target post-edited translation $\mathbf{y}$. Crucially, when a source side constraint $\check{\boldsymbol{x}}^{(i)}$ matches a sub-sequence in $\mathbf{x}$, it is required that the sub-sequence be translated as $\check{\boldsymbol{y}}^{(i)}$. See Figure 1 for an example.

4 Models

While there are existing models to address the APE task, and the lexical constrained MT task, it is not clear how to represent lexical constraints for APE models which, unlike MT models, take two sequences as input. We propose several techniques to incorporate constraints as additional inputs to the APE encoder by combining the input sequence modification used in constrained MT (Dinu et al., 2019) with the MST method of (Tebbifakhr et al., 2018). For decoding, we experiment with both the AT and the LevT decoders. The LevT decoder can additionally take advantage of different decoder initialization strategies for constrained decoding.

We first briefly show how we encode terminology constraints in the input sequence, before describing how they are incorporated into the MST and LevT APE models specifically.

Model	Input	Init.
MST	$\mathbf{x}, \boldsymbol{\gamma}$	–
MST Append	$\mathbf{x}^+, \boldsymbol{\gamma}$	–
MST Replace	$\mathbf{x}^-, \boldsymbol{\gamma}$	–
LevT	$\mathbf{x}$	$\boldsymbol{\gamma}$
LevT Append	$\mathbf{x}^+$	$\boldsymbol{\gamma}$
LevT Replace	$\mathbf{x}^-$	$\boldsymbol{\gamma}$
MS LevT	$\mathbf{x}, \boldsymbol{\gamma}$	$\check{\boldsymbol{y}}^1, \ldots, \check{\boldsymbol{y}}^{(t)}$

Figure 2: Setup for the models by the input and initialization at inference.

4.1 Encoding Lexical Constraints for APE in the Input Sequence

In the APE setting, the input to the model is the source language sentence $\mathbf{x}$ and its initial MT translation $\boldsymbol{\gamma}$. We also need to represent in $\mathbf{x}$ the translation constraints $\mathcal{C}$.

For clarity, we describe the case of representing a single translation constraint $(\check{\boldsymbol{x}}, \check{\boldsymbol{y}})$ where $\check{\boldsymbol{x}} = [\check{x}_1, \ldots, \check{x}_j]$ is a source language constraint and $\check{\boldsymbol{y}} = [\check{y}_1, \ldots, \check{y}_k]$ is its target language translation. Our approach trivially generalizes to multiple constraints. We represent the constraint $(\check{\boldsymbol{x}}, \check{\boldsymbol{y}})$ in $\mathbf{x}$ in one of two ways. Either by appending the target language constraint $\check{\boldsymbol{y}}$ after the occurrence of $\check{\boldsymbol{x}}$ in the input sequence, or by replacing occurrences of $\check{\boldsymbol{x}}$ in $\mathbf{x}$ with $\check{\boldsymbol{y}}$.

For example, if we had the constraint, $(\check{\boldsymbol{x}}, \check{\boldsymbol{y}}) = ([\check{x}_1, \check{x}_2], [\check{y}_1])$ and the source input $\mathbf{x} = [x_1, x_2, x_3, x_4]$, with $[\check{x}_1, \check{x}_2] = [x_2, x_3]$, we would obtain the following input sequences for the append

and replace methods:

- *(Append)* $\mathbf{x}^{+} = [x_1, x_2, x_3, \breve{y}_1, x_4]$
- *(Replace)* $\mathbf{x}^{-} = [x_1, \breve{y}_1, x_4]$.

To further differentiate the constraint terms from other tokens in the source sentence, a "source factor" is associated with each input token. The source factor is equal to 1 or 2 to indicate a source or target side terminology constraint, while 0 indicates an unconstrained source token. For the above examples, we would obtain the following source factors:

- *(Append)* $\mathbf{s}^{+} = [0, 1, 1, 2, 0]$
- *(Replace)* $\mathbf{s}^{-} = [0, 2, 0]$.

The source input sequence and source factor sequence are separately embedded and concatenated before they are fed into the encoder. See Figure 1 for examples of the *append* and *replace* methods applied to a source sentence.

We now describe how we use these modified input sequences in the MST and LevT models. See Figure 2 for an overview of the the proposed models and their configuration.

4.2 Multi-source Transformer

The input to an APE model is a pair of sequences, the source sentence and the MT output to be post-edited. To accommodate these two sequences, we use the MST model of Tebbifakhr et al. (2018), which uses a separate Transformer to encode each sequence. The outputs of each encoder are concatenated and attended to by the decoder.

We augment the encoder for the source sentence with the *append* and *replace* methods. Figure 1 shows an example of the inputs for the *append* and *replace* methods, $\mathbf{x}^{+}$ and $\mathbf{x}^{-}$ respectively. To account for the additional input of MT, $\boldsymbol{\gamma}$, for the source factors, we use 3 for each token in $\boldsymbol{\gamma}$. For Byte-Pair Encoding (BPE) (Sennrich et al., 2016), the corresponding source factor token is applied for all subword units.

We train three variants based on MST: an unconstrained version as the baseline (MST), and two constrained versions using the *append* (MST Append) and *replace* (MST Replace) methods as described in subsection 4.1.

4.3 Levenshtein Transformer

The LevT follows the Transformer encoder-decoder architecture. However, instead of a regular Transformer decoder, the model uses three consecutive layers to simulate the edit operations. The first layer predicts whether each token should be deleted or kept. The second layer predicts how many placeholder tokens to insert between every two consecutive tokens. The final layer then predicts the actual target token for each placeholder.

One benefit of using the LevT is its ability to initialize the decoding process with an arbitrary sequence. The first iteration of the decoding process is typically initialized with $\mathbf{y^0} = [<s>, </s>]$, but it is possible to initialize it with MT (i.e. $\boldsymbol{\gamma}$) and allow it to be subsequently refined.

Since LevT retains the single encoder and decoder structure, the changes to incorporate lexical constraints are straightforward; we apply the *append* and *replace* methods to the encoder input.

We also try augmenting the LevT similarly to the MST. Here, we have two encoders for the source, $\mathbf{x}$, and MT, $\boldsymbol{\gamma}$, respectively. During inference, we initialize the decoding string with target-side constraint terms, $\breve{\boldsymbol{y}}$, similar to the constrained decoding setup in Susanto et al. (2020).

For multiple constraints, we sort the target side terms $\breve{\boldsymbol{y}}^{(i)}$ by the order of the occurrence of $\breve{\boldsymbol{x}}^{(i)}$ in the source $\mathbf{x}$. When source and target word order diverge, we hope that the model will learn to reorder constraints correctly, but leave experimentation with constraint ordering for future work.

We train four variants of the LevT model. An unconstrained baseline model (LevT), and two constrained variants, with the same architecture as the base LevT, that incorporate constraints in the source using the *append* $(\mathbf{x}^{+})$ (LevT Append) and *replace* $(\mathbf{x}^{-})$ (LevT Replace) methods described in subsection 4.1. The decoder initialization for these models is the MT sentence, $\boldsymbol{\gamma}$, that needs to be edited. The final variant has a multi-source encoder, where $\mathbf{x}$ and $\boldsymbol{\gamma}$ are fed into separate encoders. The decoder in this case is initialized with the target sequence of the terminology constraint(s).

5 Data

5.1 APE Datasets

We use two standard English-to-German APE benchmark datasets, WMT18 PBMT (Chatterjee et al., 2018) and WMT19 NMT (Chatterjee et al., 2019). Both datasets are in the IT domain. Each example from these datasets consists of three sequences: (1) the source sentence $\mathbf{x}$, (2) its MT output $\boldsymbol{\gamma}$, and (3) its post-edited target $\mathbf{y}$.

Dataset		# of Triplets			Term%	TER	BLEU
		Train	Valid	Test			
PBMT	artificial 4M	4,390,180	1,000	-	-	-	-
	artificial 500K	526,368	-	-	-	-	-
	WMT'18 APE	24,000	2,000	2,000	88.83	24.57	62.39
NMT	eSCAPE NMT	4,999,102	1,000	-	-	-	-
	WMT'19 APE	13,442	1,000	1,023	89.52	16.92	74.60

Table 1: Statistics for data used. Term%, TER, and BLEU are provided for do-nothing case of test set.

Since the official collections are relatively small, we augment them with large synthetic datasets for pretraining: artificial (Junczys-Dowmunt and Grundkiewicz, 2016) and eSCAPE (Negri et al., 2018). The artificial dataset is generated using round-trip translation of two PBMT systems. It is already cleaned and tokenized. The eSCAPE dataset, containing 7,258,533 triplets, is created using NMT generated output from various parallel corpora. The data for eSCAPE is noisy, and we follow Lee et al. (2019)'s procedure to filter the dataset, which results in around 5 million triplets. We then tokenize the filtered data using Moses (Koehn et al., 2007).[1] For pretraining on the synthetic corpora, we set aside 1,000 randomly sampled triplets as our validation set. Table 1 summarizes the statistics of both the evaluation and pretraining datasets.

For both tasks, we use the same preprocessing steps. After tokenization, we truecase the data using Moses. We then use BPE with 32,000 merge operations on the joined vocabulary of source and target language.

5.2 Terminology Dataset

We create terminology sets for each APE dataset using Wiktionary.[2] We follow the procedure of Dinu et al. (2019), finding term translation pairs $(\check{\boldsymbol{x}}, \check{\boldsymbol{y}})$ in Wiktionary such that $\check{\boldsymbol{x}}$ is present in the source sentence $\mathbf{x}$ and $\check{\boldsymbol{y}}$ is present in the post-edited target sentence $\mathbf{y}$. We ignore stop words that appear on the source and target side. To include more morphological variations, we include matches on stemmed versions of $\check{\boldsymbol{x}}$ and $\check{\boldsymbol{y}}$ using Snowball stemming.[3] We recover the unstemmed words from the pairs to be included in the terminology dataset. In order for the model to perform the APE task well when no constraints are supplied, we keep only 25% of matched terminology constraints (i.e. we remove 75% of constraints at random).

We split the terminology dataset into training and test sets so that terminology constraints provided at test time are not seen during training. We only use the training set for the training corpora of APE datasets, and use the test sets of the terminology on the validation and test set of the APE datasets. See Table 2 for statistics of terminology coverage on the training, validation, and test splits.

With the given MT system, we can evaluate on terminology percentage for the do-nothing case, which is shown in Table 1. The original MT model already achieves a high term percentage of around 90% for PBMT and NMT tasks.

6 Experiments

We use the FAIRSEQ toolkit (Ott et al., 2019) for implementing the MST and extending the LevT.[4] We evaluate the models on translation error rate (TER) (Snover et al., 2006) and BLEU (Papineni et al., 2002) using the official evaluation script[5] for analyzing the post-editing performance. We also compute the percentage of target language term constraints present in the output (Term %) to measure the performance of the constrained models.

6.1 Constrained MT-to-APE Cascades

In our first experiment, we attempt to demonstrate the utility of constrained APE when applied to constrained MT. That is, we have some terminology constraints that we want to preserve throughout the application of MT and subsequent APE. We

[1] www.statmt.org/moses/

[2] We use the latest dump as of 06/18/2020

[3] www.nltk.org/_modules/nltk/stem/snowball.html

[4] Our code is publicly available at https://github.com/zerocstaker/constrained_ape.

[5] www.dropbox.com/s/5jw5maariwey080/Evaluation_Script.tar.gz?dl=0

Dataset		# of Triplets with Term.			Avg # of Term.		
		Train	Valid	Test	Train	Valid	Test
PBMT	artificial 4M	1,605,075	345	-	1.25	1.25	-
	artificial 500K	207,225	-	-	1.27	-	-
	WMT'18 APE	6,037	834	528	1.15	1.24	1.34
NMT	escape NMT	1,768,587	335	-	1.28	1.30	-
	WMT'19 APE	3,450	262	408	1.16	1.14	1.25

Table 2: The number of training/validation/test instances that have at least one terminology constraint and the average number of terminology constraints for those instances.

conjecture that unconstrained APE applied on top of constrained MT will potentially discard or re-translate previously translated constraints.

We experiment with all possible pipelines of MT to APE, i.e. the product of $\{\text{MT}, \text{Const. MT}\} \times \{\text{No APE}, \text{APE}, \text{Const. APE}\}$ with six total pipelines possible.

MT Models To obtain MT models for this experiment we train both a constrained and unconstrained AT MT model using the default FAIRSEQ Transformer hyperparameters and use the embedding size of 16 for the source factor embedding. We follow the settings of Dinu et al. (2019), training an unconstrained transformer and a constrained model with append method to perform English-to-German translation using the Europarl and News Commentary data, and using the WMT 2013/2017 test set as validation and test set respectively. The preprocessing steps follows that of the APE datasets.

For the constrained MT model we used the *append* input modification method to make the model constraint aware. Terminology constraints are generated according to the method described in subsection 5.2. Dinu et al. (2019) also released their Wiktionary terminology set (Wikt975)[6] and we also show evaluation results using this terminology collection. We report BLEU and terminology coverage (Term %) for our MT models on the WMT 2017 test set in Table 3.

APE Models We use the MST and the append method as the unconstrained APE and constrained APE respectively. We evaluate the MT to APE pipelines using the WMT'19 APE test set, replacing the provided MT in the triplet with the outputs from our unconstrained or constrained MT models.

[6] https://github.com/mtresearcher/terminology_dataset

MT	Our Term.		Wikt 975	
	Term%	BLEU	Term%	BLEU
AT	71.70	23.76	74.78	24.00
AT App.	93.62	24.62	93.07	24.14

Table 3: Translation result of vanilla and lexically constrained translation.

We train the APE models using the eSCAPE corpus, where 1,000 triplets are used as validation set. We use the default FAIRSEQ Transformer hyperparameters. For the constrained APE, we use the embedding size of 16 for the source factor tokens.

6.2 Benchmark APE Tasks

The APE models are trained in two step fashion. First, a general APE system is trained using a synthetic dataset until convergence. Then the model is refined on the official dataset. For the PBMT task, we follow the training procedure of Gu et al. (2019). The model is pretrained on the artificial 4M dataset, and fine-tuned on the joined dataset of the 500K artificial dataset and the 10 times up-sampled official PBMT data. For the NMT task, we pretrain on eSCAPE and fine-tune on the official NMT data.

We use the default Transformer parameters for the MST variants, with an embedding size of 16 for source factors of the constrained APE models. For the LevT models, we follow the same setup and hyper-parameters as described in Gu et al. (2019).

We compare our models to the do-nothing case, where the output of the MT, γ, is treated as the predicted post-edited sentence $\hat{\mathbf{y}}$. The unconstrained variant also serve as a basis for comparing the performance of the constrainted APE models. We also compare our models to the winning system for the tasks, MS_UEdin (Junczys-Dowmunt and Grund-

Pipeline	Term%↑	TER↓	BLEU↑
MT	45.33	70.78	15.28
cMT	86.33	70.24	15.47
MT → APE	55.35	59.56	22.87
cMT → APE	77.22	59.78	23.03
MT → cAPE	80.18	**58.70**	**23.95**
cMT → cAPE	**88.38**	59.77	23.08

Table 4: Result of different combinations of MT and APE systems. Constrained MT and APE are indicated cMT and cAPE respectively.

kiewicz, 2018) for PBMT 2018 and Unbabel_BERT Lopes et al. (2019) for NMT 2019.

7 Results and Discussions

7.1 Constrained MT-to-APE Cascade

Table 4 shows the result of the various combinations of MT and APE systems. Since the MT system is trained on news/parliamentary proceedings and not on the IT domain of the APE data, the translation quality is relatively low. Nevertheless, the constrained MT can include almost twice as many terminologies as the original model. Both APE systems improve the quality of the MT outputs, with constrained APE performing slightly better. However, constrained APE excels at including terminologies, as it consistently increases the terminology percentage from the previous MT output. When supplied with a constrained MT, the vanilla APE actually decreases the percentage of correct terminologies by 9% (86.33% to 77.22%), whereas the constrained APE model can increase it by 2% (86.33% to 88.38%).

7.2 Benchmark APE on PBMT Output

Table 5 shows the results on the PBMT task. All MST variants improve from the do-nothing case, where the output is unchanged, i.e.$\gamma = \hat{\mathbf{y}}$. Using either the *append* or *replace* methods shows similar improvements in Term%, increasing about 7% points absolutely over the do nothing case. The terminology aware MST models also see small decreases in TER and small increases in BLEU relative to the unconstrained MST model. These results are encouraging as it shows that introducing terminology constraints does not interfere with the APE system's ability to fix systematic errors.

We were unable to reproduce the result by Gu et al. (2019); we see only small improvements with

Models	Term%↑	TER↓	BLEU↑
Do-nothing	88.48	24.25	62.99
MS_UEdin	88.70	**18.01**	**72.52**
MST	90.11	19.34	70.44
MST Append	95.54	18.97	70.63
MST Replace	95.43	19.17	70.34
LevT	90.76	24.21	63.47
LevT App.	90.98	23.88	64.97
LevT Rep.	91.41	23.94	64.96
MS LevT	**97.50**	20.39	68.57

Table 5: Results for PBMT 2018.

models	Term%↑	TER↓	BLEU↑
Do-nothing	90.22	16.84	74.73
Unbabel_BERT	89.98	**16.06**	**75.96**
MST	90.66	16.46	75.61
MST Append	94.08	16.62	75.16
MST Replace	94.08	16.56	75.39
LevT	90.41	17.28	74.17
LevT App.	91.59	17.32	74.25
LevT Rep.	90.61	17.14	74.46
MS LevT	**98.04**	17.71	73.64

Table 6: Results for NMT 2019.

the LevT models relative to the do-nothing case. Additionally, the *append* and *replace* variants yield only small increases in Term% but are around 4-5% points behind the equivalent MST model. The MS LevT, however, achieves the highest terminology percentage of all models, while slightly underperforming the MST models on TER and BLEU.

None of our proposed models beat the SOTA baseline for this task on TER or BLEU, but our best model on TER, MST Append, is less than 1 percentage point worse in TER. At the same time, MST Append successfully translates 6.8% more terminology constraints than the SOTA baseline.

7.3 Benchmark APE on NMT Output

Table 6 shows the result of the NMT task. This is a more difficult post editing task as the machine translated text from NMT systems is of a higher quality than PBMT systems, and the official training corpus is smaller than that of PBMT APE (Chatterjee et al., 2018). As further evidence of the difficulty of this task, the winning system of the WMT2019

Original	Source ($\mathbf{x}$)	increasing the **magnification** can also make reshaping easier and more accurate .
	MT ($\boldsymbol{\gamma}$)	durch das Vergrößern der **Vergrößerung** können Sie außerdem das Umformen von Formen und präziser steuern .
	Post-Edit ($\mathbf{y}$)	durch das Vergrößern der **Vergrößerung** können Sie außerdem das Umformen von Formen präziser steuern .
	MST Append	Durch das Vergrößern der **Vergrößerung** können Sie außerdem das Umformen von Formen erleichtern und präziser steuern .
	MS LevT	Durch die zunehmende **Vergrößerung** können Sie außerdem das Umformen von Formen und präziser steuern .
Synonym	Post-Edit ($\mathbf{y}$)	durch das Vergrößern der **Magnifizierung** können Sie außerdem das Umformen von Formen präziser steuern .
	MST Append	durch das Vergrößern der **Magnifizierung** können Sie außerdem das Umformen von Formen vereinfachen und präziser steuern .
	MS LevT	eine Erhöhung der **Magnifizierung** kann außerdem das Umformen von Formen und präziser erleichtern .
Antonym	Post-Edit ($\mathbf{y}$)	durch das Vergrößern der **Verkleinerung** können Sie außerdem das Umformen von Formen präziser steuern .
	MST Append	durch das Vergrößern der **Vergrößerung** können Sie außerdem das Umformen von Formen vereinfachen und präziser steuern .
	MS LevT	durch die zunehmende **Verkleinerung** können Sie außerdem das Umformen von Formen und präziser steuern .

Figure 3: Example of the outputs by the MST Append and MS LevT when a synonym and an antonym is supplied in place of the original terminology pair (*magnification* - *Vergrößerung*). The synonym *Magnifizierung* (magnification) and antonym *Verkleinerung* (diminishment) is used.

Source ($\mathbf{x}$)	if you use the Image Processor , you can **save** the files directly to JPEG format in the size that you want them .
Post-Edit ($\mathbf{y}$)	wenn Sie den Bildprozessor verwenden , können Sie die Dateien direkt im JPEG-Format in der gewünschten Größe **speichern**
Synonym	wenn Sie den Bildprozessor verwenden , können Sie die Dateien direkt im JPEG-Format in der gewünschten Größe **sichern**
Antonym	wenn Sie den Bildprozessor verwenden , können Sie die Dateien direkt im JPEG-Format in der gewünschten Größe **löschen**

Figure 4: Example of data augmentation. The original term pair is (*save*, *speichern*). We replace the target terminology *speichern* with the synonym *sichern* (to store for future use) or the antonym *löschen* (to delete).

APE shared task is able to achieve a mere 0.78 point decrease in TER.

The two terminology-aware MST models (*append* and *replace*) are able to improve Term% over the baseline, at the cost of a slight increase in TER and decrease in BLEU, but both are better than doing nothing. The LevT and its variants perform worse than doing nothing in terms of TER and BLEU, but has a small gain in Term%. The MS LevT again achieves the highest Term% but does worse than the do-nothing case on TER and BLEU.

8 Analyzing Constraint Translation Behavior

Terminology constrained APE aims to add some degree of user control over the APE process without destabilizing the general post-editing behavior of the decoder. However, the imposition of rare or unusual terminology constraints will necessarily be in conflict with the decoder language model, which will give higher probabilities to terminology translations found frequently in the training data.

In practice, a user may specify a terminology constraint that is not well represented in the training distribution. For example, a user may want a product description translated using location specific brand names or marketing copy. Ideally, a terminology constrained model would reliably produce these terms and use them appropriately even if they do not rank highly by the decoder.

Additionally, it is desirable that the addition of terminology constraints does not lead to large changes in the model's output. Since terminology constraints are only bound to a word or phrase, the model should only need to make minimal changes between the unconstrained and constrained output. Large changes in output may make it harder for a user to anticipate the effects of a constraint which may make constrained APE less useful in practice.

	WMT'19 APE			Augmentation		
	Term%↑	TER↓	BLEU↑	Term%↑	TER↓	BLEU↑
Do-nothing	90.22	16.84	74.73	1.66	24.77	62.56
MST Append	94.08	16.62	75.16	7.47	24.92	61.80
MST Append + *pretrain*	94.08	16.46	75.25	18.67	23.70	64.38
MST Append + *pretrain* + *ft*	93.85	**16.29**	**75.38**	43.15	**21.85**	**67.41**
MS LevT	98.04	17.71	73.64	43.57	33.07	54.33
MS LevT + *pretrain*	**99.09**	17.18	74.22	52.70	29.79	60.24
MS LevT + *pretrain* + *ft*	98.41	17.00	74.66	**63.07**	29.66	60.47

Table 7: Results with data augmentation for the official APE data, as well on the augmented dataset consisting of synonyms and antonyms generated from Wiktionary. The size of the additional data for test set is 236.

	Synonym			Antonyms		
	Term%↑	TER↓	BLEU↑	Term%↑	TER↓	BLEU↑
MST Append	7.33	**1.31**	**97.80**	8.88	**1.06**	**98.01**
MST Append + *pretrain*	16.75	2.81	94.19	28.88	3.81	92.94
MST Append + *pretrain* + *ft*	38.74	5.48	88.86	66.66	6.46	87.50
MS LevT	41.36	19.57	70.60	57.77	17.56	74.25
MS LevT + *pretrain*	47.12	18.33	72.27	82.22	13.28	79.42
MS LevT + *pretrain* + *ft*	**43.97**	18.25	73.25	**77.78**	11.99	80.41

Table 8: Structural change from the output of constrained models using the correct terminology. We split the dataset by synonyms and anotnyms, consisting of 191 and 45 samples respectively.

We refer to this behavior as *systematic copying*, i.e. the model should behave in a transparent and stable way, enforcing terminology constraints even when they strongly disagree with the decoder language model, while only making minimally necessary changes in the output to do so.

By harvesting terminology constraints from the training data, we run the risk that the model simply learns to draw some translation hints from the supplied terminology constraints, but does not actually learn this systematic copying behavior. That is, it never truly sees an out-of-sample constraint that is extremely unlikely from the perspective of the decoder language model.

To test whether our proposed models indeed learn this systematic copying behavior we perform a qualitative experiment, comparing model outputs when supplying different constraints for a source word, by varying whether the target language constraint was (a) the original target language constraint specified in the test set, (b) a target language synonym of the original constraint term, (c) a target language antonym of the original constraint term, or (d) a totally random term in the target language.

While antonym and random term constraints might not seem to correspond to realistic use cases, they let us to examine the effects of specifying a constraint where the source and target language terms are extremely semantically divergent. Additionally, they simulate scenarios where translations for names vary dramatically by region. For example, the cleaning product called "Mr. Clean" in the U.S. is called "Meister Proper" in Germany.

As can be seen in Figure 3, our qualitative exploration reveals that synonym, antonym, and random terminology constraints are frequently not included in the output. For example the MST model fails to generate the antonym *Verkleinerung*. This suggests that target side constraints that are unseen during training may be ignored by the model, and that the models are not learning to systematically copy arbitrary constraints.

8.1 Data Augmentation Experiment

The results of our qualitative exploration suggests that the model would benefit from seeing more semantically divergent terminology constraints during training. To that end, we propose a data aug-

mentation experiment to increase the robustness of the APE models. We create novel training instances by replacing the target language term constraint with either a synonym or antonym (using Wiktionary), as well as replacing it's occurrence in the post-edited target translation. This results in 837,127 additional samples for eSCAPE corpus, and 2587 additional samples for official NMT data. See Figure 4 for examples.

We then train MST append and MS LevT with the augmented pretraining corpus. We experiment with using augmented data only for pretraining (*pretrain*) and for the fine-tuning process (*ft*). The result can be seen in Table 7.

Interestingly, on the WMT19 test set, the data augmentation helps with TER and BLEU while having only a slight effect on Term%. For the LevT, data augmentation helps all metrics.

Since the WMT19 APE test data contains few unusual constraints, the effect of the augmented data is relatively small. When we create antonym and synonyms examples from the WMT19 APE test data, we see fairly positive trends, with pretraining and fine-tuning yielding additive reductions in TER and gains in BLEU. This suggests that the augmentation method has a positive effect on the systematic copying behavior of the model.

8.2 Post-Edit Stability

To quantify the stability of the APE models, we compare the constrained APE output when given a target side synonym or antonym to the output of that same model under the original test set constraint using TER and BLEU. Under this setting, higher BLEU and lower TER indicate that the model makes minimal changes when inserting a semantically divergent constraint. We also report Term% to show how often the terminology was correctly translated given the input. We refer to a model with high Term% and BLEU but low TER as a stable model. Results of this experiment are shown in Table 8.

There are several takeaways from this experiment. First, the LevT TER scores are higher on average than the MST model suggesting that the LevT model is less stable, producing different translations for each target side constraint change.

Second, as sensitivity to constraints increases (i.e. Term% goes up), TER generally goes up, implying that models make more structural changes to the overall output in order to accommodate constraints. Future work on refinement tasks like APE may benefit from including an explicit objective function to encourage output stability.

Finally, synonyms are harder to translate than antonyms (i.e. Synonym Term% is lower than Antonym Term% for all models/training configurations). This may be because the original target side constraints are better represented in the decoder language model and are likely have higher probability than a synonym when either could be plausibly used in the same context. Antonyms may be less likely and therefore easier to override the preferences of decoder.

9 Conclusion and Future Work

This work introduces the terminology constrained APE task and several MST and LevT model variants for incorporating lexical constraints during post-editing. Furthermore, we show that constrained APE is necessary for preserving lexical constraints in a MT to APE pipeline. Evaluations on standard APE benchmarks show that terminology constraints are satisfied while improving the original MT quality. Finally, we show that the constrained APE models do not learn a robust systematic copying behavior, and propose a data augmentation method to help mitigate this issue. In future work, we hope to explore ways of modifying model architecture or training algorithms to further improve the systematic copying behavior.

Acknowledgments

This research is based upon work supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via contract #FA8650-17-C-9117. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

References

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Raphael Rubino, Lucia Specia, and Marco Turchi. 2017. Findings of the 2017 Conference on Machine Translation (WMT17). In *Proceedings of the Second Conference on Machine Translation*, pages 169–214, Copenhagen, Denmark. Association for Computational Linguistics.

Rajen Chatterjee, Christian Federmann, Matteo Negri, and Marco Turchi. 2019. Findings of the WMT 2019 Shared Task on Automatic Post-Editing. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 13–30, Florence, Italy. Association for Computational Linguistics.

Rajen Chatterjee, Matteo Negri, Raphael Rubino, and Marco Turchi. 2018. Findings of the WMT 2018 Shared Task on Automatic Post-Editing. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 710–725, Belgium, Brussels. Association for Computational Linguistics.

Georgiana Dinu, Prashant Mathur, Marcello Federico, and Yaser Al-Onaizan. 2019. Training Neural Machine Translation to Apply Terminology Constraints. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3063–3068, Florence, Italy. Association for Computational Linguistics.

Jiatao Gu, Changhan Wang, and Junbo Zhao. 2019. Levenshtein Transformer. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 11181–11191. Curran Associates, Inc.

Chris Hokamp and Qun Liu. 2017. Lexically Constrained Decoding for Sequence Generation Using Grid Beam Search. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1535–1546, Vancouver, Canada. Association for Computational Linguistics.

Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2016. Log-linear Combinations of Monolingual and Bilingual Neural Machine Translation Models for Automatic Post-Editing. In *Proceedings of the First Conference on Machine Translation*, pages 751–758, Berlin, Germany. Association for Computational Linguistics.

Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2018. MS-UEdin Submission to the WMT2018 APE Shared Task: Dual-Source Transformer for Automatic Post-Editing. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 835–839, Belgium, Brussels. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

WonKee Lee, Jaehun Shin, and Jong-Hyeok Lee. 2019. Transformer-based Automatic Post-Editing Model with Joint Encoder and Multi-source Attention of Decoder. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 112–117, Florence, Italy. Association for Computational Linguistics.

António V. Lopes, M. Amin Farajian, Gonçalo M. Correia, Jonay Trénous, and André F. T. Martins. 2019. Unbabel's Submission to the WMT2019 APE Shared Task: BERT-Based Encoder-Decoder for Automatic Post-Editing. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 118–123, Florence, Italy. Association for Computational Linguistics.

Matteo Negri, Marco Turchi, Rajen Chatterjee, and Nicola Bertoldi. 2018. ESCAPE: a Large-scale Synthetic Corpus for Automatic Post-Editing. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A Fast, Extensible Toolkit for Sequence Modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Matt Post and David Vilar. 2018. Fast Lexically Constrained Decoding with Dynamic Beam Allocation for Neural Machine Translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1314–1324, New Orleans, Louisiana. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Michel Simard, Cyril Goutte, and Pierre Isabelle. 2007. Statistical Phrase-Based Post-Editing. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 508–515, Rochester, New York. Association for Computational Linguistics.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of Association for Machine Translation in the Americas*, pages 223–231.

Raymond Hendy Susanto, Shamil Chollampatt, and Liling Tan. 2020. Lexically Constrained Neural Machine Translation with Levenshtein Transformer. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3536–3543, Online. Association for Computational Linguistics.

Amirhossein Tebbifakhr, Ruchit Agrawal, Matteo Negri, and Marco Turchi. 2018. Multi-source Transformer with Combined Losses for Automatic Post Editing. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 846–852, Belgium, Brussels. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Author Index