IIT Bombay's WMT22 Automatic Post-Editing Shared Task Submission

Sourabh Deoghare and **Pushpak Bhattacharyya**
Computation for Indian Langauge Technology (CFILT)
IIT Bombay, India
{sourabhdeoghare, pb}@cse.iitb.ac.in

Abstract

This paper describes IIT Bombay's submission to the WMT-22 Automatic Post-Editing (APE) shared task for the English-Marathi (En-Mr) language pair. We follow the curriculum training strategy to train our APE system. First, we train an encoder-decoder model to perform translation from English to Marathi. Next, we add another encoder to the model and train the resulting *dual-encoder single-decoder* model for the APE task. This involves training the model using the synthetic APE data in multiple training stages and then fine-tuning it using the real APE data. We use the LaBSE technique to ensure the quality of the synthetic APE data. For data augmentation, along with using candidates obtained from an external machine translation (MT) system, we also use the phrase-level APE triplets generated using phrase table injection. As APE systems are prone to the problem of 'over-correction', we use a sentence-level quality estimation (QE) system to select the final output between an original translation and the corresponding output generated by the APE model. Our approach improves the TER and BLEU scores on the development set by -3.92 and +4.36 points, respectively. Also, the final results on the test set show that our APE system outperforms the baseline system by -3.49 TER points and +5.37 BLEU points.

1 Introduction

Automatic Post-Editing (APE) is a post-processing task in a Machine Translation (MT) workflow. It aims to automatically identify and correct errors in MT outputs (Chatterjee et al., 2020). Läubli et al. (2013) and Pal et al. (2016) show that APE systems have the potential to reduce human effort by automatically correcting repetitive translation errors.

The initial years of the WMT APE shared task focused on correcting errors in Statistical Machine Translation (SMT) translations, where participants explored various statistical and neural APE approaches (Bojar et al., 2017). Although neural APE approaches showed high potential for significantly improving the quality of SMT translations, these approaches faced challenges in improving translations obtained from relatively-more-robust neural machine translation (NMT) systems (Chatterjee et al., 2018). A possible reason for this could be that correcting a high-quality translation requires fewer edits, and therefore APE approaches need to be precise in identifying and in correcting the errors. Also, the neural APE approaches use large neural networks that require significant training data. APE training data consists of 'triplets' in the form of source sentence (*src*), its translation generated using an MT system (*mt*), and a human post-edited version of the translation (*pe*). Obtaining *pe* is an expensive task in terms of time and money; therefore, there is a lack of large APE datasets.

To deal with this problem, various data augmentation techniques have been proposed (Junczys-Dowmunt and Grundkiewicz, 2016; Negri et al., 2018; Lee et al., 2020b). Wang et al. (2020) used imitation learning to filter the APE data for tackling the distributional difference between real and synthetic APE data. Wei et al. (2020) augmented the APE training data with translations generated using a different MT system. Inspiring from the work of Sen et al. (2021), we augment the APE data by generating phrase-level APE triplets using SMT phrase tables. To ensure the quality of the synthetic data, we use the LaBSE technique (Feng et al., 2022) and filter low-quality triplets.

Another effective approach for dealing with the problem of data sparsity is transfer learning in which pre-trained models are adapted to the APE task (Lopes et al., 2019). An APE system needs to understand both the source and target languages to obtain joint encoding of *src* and *mt*. Therefore, Lee et al. (2020a) uses a cross-lingual language model instead of a monolingual one. Unlike

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 682–688
Abu Dhabi, December 7–8, 2022. ©2022 Association for Computational Linguistics

these approaches, Wei et al. (2020); Sharma et al. (2021) use a pre-trained NMT model and adapts it to the APE task. Oh et al. (2021) has proposed the Curriculum Training Strategy (CTS) that gradually adapts pre-trained models to the APE task.

Although recent APE systems use a single encoder to encode both the source sentence and its translation (Oh et al., 2021; Lee et al., 2020a), we use separate encoders for encoding *src* and *mt* as English and Marathi do not share much vocabulary; and belong to different language families. We use IndicBERT (Kakwani et al., 2020) to initialize weights of our the *src* encoder and *mt* encoder. We train and fine-tune our models using the CTS over the good-quality APE data. The training data is also augmented with external MT candidates and phrase-level APE triplets. It is known that APE systems are prone to making unnecessary edits to translation output (Chatterjee et al., 2020). To mitigate this issue of over-correction, we use a sentence-level QE system to select the final output. When evaluated on the development set, our approach improves the TER (Snover et al., 2006) by -3.92 points and the BLEU (Papineni et al., 2002) by +4.98 points. Similarly, the final results on the test set show that our APE system outperforms the baseline system by -3.49 TER points and +5.37 BLEU points. We summarize the main features of our approach as follows:

- We use two separate encoders to generate representations for *src* and *mt*. We also use the IndicBERT language model to initialize the weights for both our encoders.

- We filter low-quality APE triplets from the synthetic data using LaBSE-based filtering.

- We divide the APE training step using CTS into two phases. We train the APE model in the first phase using out-of-domain synthetic APE data. In the next phase, we train the APE model using only the in-domain APE data.

- We follow two approaches for data augmentation: (1) As per the recent trend, we use external MT candidates. (2) We generate phrase-level APE triplets using SMT phrase tables.

- APE systems are prone to the problem of over-correction. Therefore, we use a sentence-QE system to select the final output between the APE output and the original translation.

2 Approach

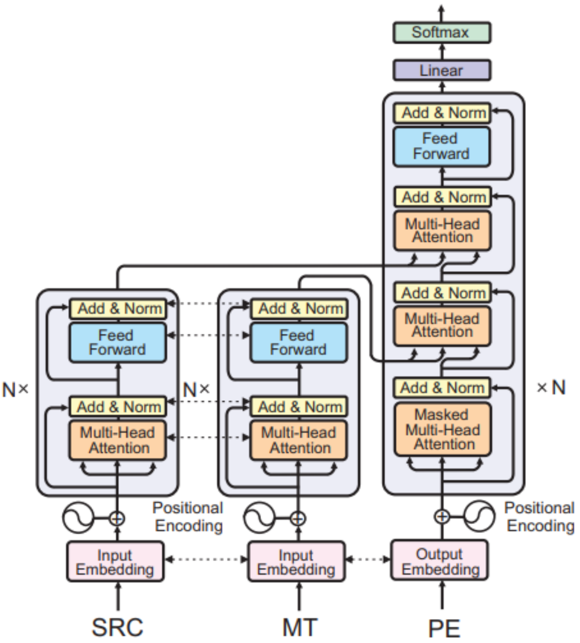

Figure 1: Dual-encoder Single Decoder Architecture. Dashed arrows represent tied parameters and common embedding matrices for encoders and decoder.(Junczys-Dowmunt and Grundkiewicz, 2018)

Our APE model is based on the transformer (Vaswani et al., 2017) architecture. Figure 1 shows the architecture of our APE model. In this section, we discuss the details of our approach.

2.1 Dual-Encoder Single-Decoder APE Model

The APE task is usually treated as an NMT-like task. Recent approaches use a single encoder to encode a source sentence and its translation (Oh et al., 2021; Lee et al., 2020a). Such an approach may work well when the source and target languages share the vocabulary (Kanojia et al., 2021). However, for English and Marathi, there is no vocabulary overlap, and also, the script used in both languages is different (Kanojia et al., 2020). Therefore, for developing an English-Marathi APE system, we use two separate encoders to encode *src* and *mt* (Junczys-Dowmunt and Grundkiewicz, 2018).

We apply transfer learning by using IndicBERT to initialize weights of the *src encoder* and the *mt encoder*. We choose IndicBERT as it is trained over text in Indian languages and English. We use a single transformer-based decoder that attends to representations of both *src* and *mt* and generates a post-edited version of the *mt*. We add one more cross-attention layer above the available cross-attention layer in the decoder. We pass the representation generated by *mt encoder* to the first cross-attention layer. The newly-added cross-attention layer receives two inputs: output of the first attention layer and representation generated by the *mt encoder*. Such placement allows the decoder to first attend to *mt*, which is prone to mistakes, and then it attends to *src*, which doesn't involve any errors. We share parameters between encoders, but the encoders generate different activations, and different attention layers receive the outputs of these encoders in the decoder. During the fine-tuning phase, we concatenate *mt* and *external MT candidate* using a special token '[SEP]' and pass this concatenated sequence to the *mt encoder*.

2.2 Sentence-Level Quality Estimation

In the Sentence-level Quality Estimation (QE) task, the machine-translated sentence is evaluated by human annotators by providing each instance with a Direct Assessment (DA) score (ranging from 0 to 100). These scores are then normalized using *z-score normalization*. A source sentence and the corresponding machine-translated output are passed to the sentence-level QE (sentence-QE) system as inputs, and it predicts a z-standardized DA score denoting the quality of translation.

We use the MonoTransquest (Ranasinghe et al., 2020), a XLM-R (Conneau et al., 2020) based model to obtain representations of the inputs. The XLM-R model is trained using a 2.5TB multilingual dataset retrieved from the CommonCrawl databases, which includes 104 languages. It is trained using the RoBERTa's masked language modelling (MLM) objective (Liu et al., 2019). We use the training (18K samples), and development (1K samples) sets shared in the WMT-22 Sentence-QE English-Marathi sub-task to train our sentence-QE model.

We use this sentence-QE model to rate the original translation and the output generated by our system. We then compare the ratings for both these sequences and select the one with a higher rating as the final output.

2.3 Curriculum Training Strategy (CTS)

We follow the CTS (Oh et al., 2021) to train our APE model. It involves gradually adapting a model to more complex tasks. In the first step, we train an encoder-decoder model for performing English to Marathi translation. We then add another encoder to the encoder-decoder model and train the resulting *dual-encoder single-decoder* model for the APE task using synthetic APE data in two phases. In the first phase, we train the APE model using APE triplets belonging to any domain except the General, News, and Healthcare domains. In the second phase, we train the model using synthetic APE triplets of the General, News, and Healthcare domains. Finally, we fine-tune the APE model using in-domain real APE data and external MT candidates.

2.4 Data Augmentation

Before using the synthetic APE data during the training steps of the CTS, we filter the low-quality triplets by using the LaBSE-based filtering (Feng et al., 2022). We do this to ensure adequate quality of the synthetic APE data. To do so, we first generate embeddings of the *src* and *pe* using the LaBSE model and normalize them. Then, we compute the cosine similarity between these normalized embeddings. If the cosine similarity is less than 0.91, we discard the corresponding APE triplet. Our experimental results show the importance of using good-quality APE data to train APE systems.

We also generate the phrase-level APE triplets using the good-quality synthetic APE data and the real APE data. We follow the procedure described by Sen et al. (2021) and extend it for the phrase-level triplet injection for APE. First, we use the Moses (Koehn et al., 2007) SMT system and train *src-mt* and *src-pe* phrase-based SMT systems. We then extract these phrase pairs from both SMT systems. In the next step, we collect pairs of phrase-pairs having same *src* from the *src-mt* and *src-pe* phrase tables. Finally, we follow the steps used in the LaBSE-based filtering and get cosine similarity scores for both the phrase pairs having the same *src*. If both the scores are more than 0.91, we combine these two phrase pairs to form a triplet and add it to the APE dataset.

To generate the external MT candidates, we train an mT5 (Xue et al., 2021) based English-Marathi NMT model over a publicly available English-Marathi parallel corpora (Samanantar (Ramesh et al., 2022), Anuvaad[1], Tatoeba[2], and ILCI (Bansal et al., 2013)) of around 6M parallel sentence pairs. We use the external MT candidates during the fine-tuning phase.

[1] Anuvaad: Github Repo

[2] Tatoeba Project

System	TER$\downarrow$	BLEU$\uparrow$
Do Nothing (Baseline)	22.93	64.51
+ CTS-based Training and External MT	20.08	67.39
+ LaBSE-based Data Filtering and in-domain training data	19.73	67.86
+ Phrase-level APE triplets	19.39	68.35
+ Sentence-level QE	**19.01**	**68.87**

Table 1: Results on the WMT-22 APE Development Set.

System	TER$\downarrow$	BLEU$\uparrow$
Do Nothing (Baseline)	20.28	67.55
IIT Bombay's Submission	**16.79**	**72.92**

Table 2: Results on the WMT-22 APE Test Set.

3 Experimental Setup

3.1 Dataset

This year's APE shared task focused only on the English-Marathi language pair. The real APE training data contains 18K APE triplets, and this APE data belongs to the General, Healthcare, and Tourism domains. The organizers also shared the synthetic APE data of various domains totaling around 25M APE triplets. As participants, we were permitted to use external data for this task.

To train a translation model, we use the publicly available English-Marathi parallel corpora of size around 6M parallel sentence pairs. For data augmentation, we first generate phrase-level APE triplets using synthetic and real APE data and then randomly select 50000 phrase-level APE pairs for augmenting with the synthetic APE data and 10000 for augmenting with real APE data.

3.2 Training Hyperparameters

We used NVIDIA DGX A100 GPUs for our experiments. We trained our models with a batch size of 32. We set the number of maximum epochs to 1000 with early stopping patience of 5. We used the Adam optimizer with a learning rate of 5 x 10^{-5}, $\beta_1 = 0.9$, and $\beta_2 = 0.997$. We set the number of warmup steps to 25K. On the decoder side, We used beam search with the beam size set to 5. For the LaBSE-based filtering, we used a threshold value of 0.91 for cosine similarity to ensure that *mt* and textitpe are similar to each other.

4 Results

In Table 2, we report the results of our APE system by evaluating it on the development set. To estimate the quality of our APE system output compared to the human-generated references, we use BLEU and TER score between the APE output and *pe*. Table 2 compiles the results of our experiments performed on the development set.

We compare the results of our experiments against a 'Do Nothing' APE baseline that simply outputs *mt* without any modification. When we trained our model using CTS and external MT candidates to increase feature diversity, the TER and BLEU scores improved to 20.08 TER points and 67.39 BLEU points from the baseline TER and BLEU scores of 22.93 and 64.51, respectively. The third row in the 2 shows the results of an experiment where we use a good-quality synthetic dataset for APE training obtained by filtering low-quality triplets using LaBSE-based filtering. The experiment also involves training the APE model in two phases: first, the model is trained on out-of-domain synthetic data and then on in-domain synthetic data. This setting brings -3.2 and +3.35 TER and BLEU score improvements over the baseline, and *underlines the importance of using good-quality in-domain APE data.*

The only change we make for performing the next experiment is augmenting the synthetic and real APE data using phrase-level APE triplets. Results of this experiment show that performance improves over the baseline by -3.54 TER points and +3.84 BLEU points. Towards the end, we also used a sentence-QE system to rate the original translation and the APE output. We then select one of them with a higher rating as the final output of our APE system. With the combination of the APE model and sentence-QE system, we see that the TER score improves to 19.01 points, and BLEU score increases to 68.87 points; which *shows that using the sentence-level QE system is an effective* *approach to discard APE output, in cases of over-correction.*

As per the information received by the shared task organizers, our APE system achieves a TER score of 16.79 points and a BLEU score of 72.92 when evaluated on the official test set, which is -3.49 TER points and +5.37 BLEU points improvement over the baseline.

5 Conclusions and Future Work

This paper presents our APE system submitted to the WMT-22 APE English-Marathi Shared task. We use a dual-encoder single-decoder model where both encoders are initialized using IndicBERT. We propose a new way to generate artificial phrase-level APE triplets by extending the phrase-pair injection method used in MT for APE. We show that augmenting APE training data with these phrase-level triplets and training the model with the CTS on good-quality in-domain APE data improves the performance of the APE system. Furthermore, we also explore using the sentence-level QE system to discard low-quality APE outputs. Evaluation of our APE system shows that our approach achieves significant gains on the WMT-22 APE development and test sets.

In future, we would like to extend this approach for automatic post-editing with the help of word-level quality estimation and come up with a single architecture for performing both the QE tasks along with APE. We would also like to attempt a multilingual APE system with a shared decoder across multiple languages.

6 Limitations

We use in-domain data to train the APE model in the last training stage and the fine-tuning stage. It makes the APE system robust in post-editing in-domain translations, but it also makes it sophisticated. We observe that the system's performance worsens when we pass out-of-domain translations to the system. Similarly, we observe poor performance when translations with distributional differences from the real APE data are passed to the APE system. We use a sentence-level QE system to compare the quality of the APE output and the original translation. Even though it helps us to get rid of poor-quality APE outputs, the APE system itself does not get benefited from it.

Acknowledgements

We would like to thank the anonymous reviewers. Their insightful comments helped us in improving the current version of the paper.

References

Akanksha Bansal, Esha Banerjee, and Girish Nath Jha. 2013. Corpora creation for indian language technologies–the ilci project. In *the sixth Proceedings of Language Technology Conference (LTC '13)*.

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, 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*, pages 646–659, Online. 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.

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.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic BERT sentence embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891, Dublin, Ireland. 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: 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.

Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, Gokul N.C., Avik Bhattacharyya, Mitesh M. Khapra, and Pratyush Kumar. 2020. IndicNLPSuite: Monolingual Corpora, Evaluation Benchmarks and Pre-trained Multilingual Language Models for Indian Languages. In *Findings of EMNLP*.

Diptesh Kanojia, Raj Dabre, Shubham Dewangan, Pushpak Bhattacharyya, Gholamreza Haffari, and Malhar Kulkarni. 2020. Harnessing cross-lingual features to improve cognate detection for low-resource languages. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1384–1395, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Diptesh Kanojia, Prashant Sharma, Sayali Ghodekar, Pushpak Bhattacharyya, Gholamreza Haffari, and Malhar Kulkarni. 2021. Cognition-aware cognate detection. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3281–3292, 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, 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.

Samuel Läubli, Mark Fishel, Gary Massey, Maureen Ehrensberger-Dow, and Martin Volk. 2013. Assessing post-editing efficiency in a realistic translation environment. In *Proceedings of the 2nd Workshop on Post-editing Technology and Practice*, Nice, France.

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*, pages 777–782, Online. Association for Computational Linguistics.

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*, pages 783–788, Online. 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*, abs/1907.11692.

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).

Shinhyeok Oh, Sion Jang, Hu Xu, Shounan An, and Insoo Oh. 2021. Netmarble AI center's WMT21 automatic post-editing shared task submission. In *Proceedings of the Sixth Conference on Machine Translation*, pages 307–314, Online. Association for Computational Linguistics.

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.

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.

Gowtham Ramesh, Sumanth Doddapaneni, Aravinth Bheemaraj, Mayank Jobanputra, Raghavan AK, Ajitesh Sharma, Sujit Sahoo, Harshita Diddee, Mahalakshmi J, Divyanshu Kakwani, Navneet Kumar, Aswin Pradeep, Srihari Nagaraj, Kumar Deepak, Vivek Raghavan, Anoop Kunchukuttan, Pratyush Kumar, and Mitesh Shantadevi Khapra. 2022. Samanantar: The Largest Publicly Available Parallel Corpora Collection for 11 Indic Languages. *Transactions of the Association for Computational Linguistics*, 10:145–162.

Tharindu Ranasinghe, Constantin Orasan, and Ruslan Mitkov. 2020. TransQuest at WMT2020: Sentence-level direct assessment. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1049–1055, Online. Association for Computational Linguistics.

Sukanta Sen, Mohammed Hasanuzzaman, Asif Ekbal, Pushpak Bhattacharyya, and Andy Way. 2021. Neural machine translation of low-resource languages using smt phrase pair injection. *Natural Language Engineering*, 27(3):271–292.

Abhishek Sharma, Prabhakar Gupta, and Anil Nelakanti. 2021. Adapting neural machine translation for automatic post-editing. In *Proceedings of the Sixth Conference on Machine Translation*, pages 315–319, Online. Association for Computational Linguistics.

Matthew Snover, Bonnie Dorr, Rich 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: Technical Papers*, pages 223–231, Cambridge, Massachusetts, USA. Association for Machine Translation in the Americas.

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*, volume 30. 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*, pages 789–796, Online. 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. 2020. HW-TSC's participation in the WMT 2020 news translation shared task. In *Proceedings of the Fifth Conference on Machine Translation*, pages 293–299, Online. Association for Computational Linguistics.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.