Microsoft's Submission to the WMT2018 News Translation Task: How I Learned to Stop Worrying and Love the Data

Marcin Junczys-Dowmunt
Microsoft
1 Microsoft Way
Redmond, WA 98121, USA

Abstract

This paper describes the Microsoft submission to the WMT2018 news translation shared task. We participated in one language direction – English-German. Our system follows current best-practice and combines state-of-the-art models with new data filtering (dual conditional cross-entropy filtering) and sentence weighting methods. We trained fairly standard Transformer-big models with an updated version of Edinburgh's training scheme for WMT2017 and experimented with different filtering schemes for Paracrawl. According to automatic metrics (BLEU) we reached the highest score for this subtask with a nearly 2 BLEU point margin over the next strongest system. Based on human evaluation we ranked first among constrained systems. We believe this is mostly caused by our data filtering/weighting regime.

1 Introduction

This paper describes the Microsoft submission to the WMT2018 (Bojar et al., 2018) news translation shared task. We only participated in one language direction – English-German. Our system follows current best-practice and combines state-of-the-art models with new data filtering and weighting methods. According to automatic metrics (BLEU) we reached the highest score for this subtask with a nearly 2 BLEU point margin over the next strongest system. We believe this is mostly caused by our data filtering/weighting regime. Based on human evaluation we ranked first among constrained systems.

Our title references the fact that we built fairly standard models, updating existing baselines for WMT2017 to the new Transformer model (Vaswani et al., 2017), but spent more time on data cleaning and work with Paracrawl. As a side-effect we came up with a new parallel data filtering method which we call dual conditional cross-entropy filtering.

2 The Marian toolkit

For our experiments, we use Marian (Junczys-Dowmunt et al., 2018) an efficient Neural Machine Translation framework written in pure C++ with minimal dependencies. Microsoft Translator employees are contributing code to Marian. In the evolving eco-system of open-source NMT toolkits, Marian occupies its own niche best characterized by two aspects:

- It is written completely in C++11 and intentionally does not provide Python bindings; model code and meta-algorithms are meant to be implemented in efficient C++ code.
- It is self-contained with its own back end, which provides reverse-mode automatic differentiation based on dynamic graphs.

Marian is distributed under the MIT license and available from `https://marian-nmt.github.io` or the GitHub repository `https://github.com/marian-nmt/marian`.

3 NMT architectures

In Junczys-Dowmunt et al. (2018), we prepared a baseline setup for Marian which reproduces the highest scoring NMT system (Sennrich et al., 2017) in terms of BLEU during the WMT 2017 shared task on English-German news translation (Bojar et al., 2017). We further replaced the original RNN-based architecture with Transformer-style models from Vaswani et al. (2017) corresponding to their "base" and "big" architectures. In this section, we reuse the recipe and the proposed models as a set of strong baselines.

3.1 Deep transition RNN architecture

The model architecture in Sennrich et al. (2017) is a sequence-to-sequence model with single-layer RNNs in both, the encoder and decoder. The RNN

Proceedings of the Third Conference on Machine Translation (WMT), Volume 2: Shared Task Papers, pages 429–434
Brussels, Belgium, October 31 - November 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-6415

System	2016	2017	2018*
Deep RNN (x1)	34.3	27.7	-
+Ensemble (x4)	35.3	28.2	-
+R2L Reranking (x4)	35.9	28.7	-
Transformer-base (x1)	35.6	28.8	43.2
+Ensemble (x4)	36.4	29.4	44.0
+R2L Reranking (x4)	36.8	29.5	44.4
Transformer-big (x1)	36.6	30.0	44.2
+Ensemble (x4)	37.2	30.5	45.2
+R2L Reranking (x4)	37.6	30.7	45.5

Table 1: BLEU results for our replication of the UEdin WMT17 system for the en-de news translation task. We reproduced most steps and replaced the deep RNN model with Transformer models. Asterisk * marks post-submission evaluation.

in the encoder is bi-directional. Depth is achieved by building stacked GRU-blocks resulting in very tall RNN cells for every recurrent step (deep transitions). The encoder consists of four GRU-blocks per cell, the decoder of eight GRU-blocks with an attention mechanism placed between the first and second block. As in Sennrich et al. (2017), embeddings size is 512, RNN state size is 1024. We use layer-normalization (Ba et al., 2016) and variational drop-out with $p = 0.1$ (Gal and Ghahramani, 2016) inside GRU-blocks and attention.

3.2 Transformer architectures

We very closely follow the architecture described in Vaswani et al. (2017) and their "base" and "big" models.

3.3 Training recipe

Modeled after the description from Sennrich et al. (2017), we reuse the example available at `https://github.com/marian-nmt/marian-examples` and perform the following steps:

- preprocessing of training data, tokenization, true-casing[1], vocabulary reduction to 36,000 joint BPE subword units (Sennrich et al., 2016) with a separate tool.[2]
- training of a shallow model for back-translation on parallel WMT17 data;
- translation of 10M German monolingual news sentences to English; concatenation of artificial training corpus with original data (times two) to produce new training data;
- training of four left-to-right (L2R) deep models (either RNN-based or Transformer-based);
- training of four additional deep models with right-to-left (R2L) orientation; [3]
- ensemble-decoding with four L2R models resulting in an n-best list of 12 hypotheses per input sentence;
- rescoring of n-best list with four R2L models, all model scores are weighted equally;
- evaluation on newstest-2016 (validation set) and newstest-2017 with sacreBLEU.[4]

At this stage we did not update to WMT2018 parallel or monolingual training data. This might put us at a slight disadvantage, but we could reuse models and back-translated data that was produced earlier.

We train the deep RNN models and Transformer-base models with synchronous Adam on 8 NVIDIA Titan X Pascal GPUs with 12GB RAM for 10 epochs each. The back-translation model is trained with asynchronous Adam on 8 GPUs. The transformer-big models are trained until convergence on four NVIDIA P40 GPUs with 24GB RAM. We do not specify a batch size as Marian adjusts the batch based on available memory to maximize speed and memory usage. This guarantees that a chosen memory budget will not be exceeded during training and uses maximal batch sizes.

All models use tied embeddings between source, target and output embeddings (Press and Wolf, 2017). Contrary to Sennrich et al. (2017) or Vaswani et al. (2017), we do not average checkpoints, but maintain a continuously updated exponentially averaged model over the entire training run. Following Vaswani et al. (2017), the learning rate is set to 0.0003 (0.0002 for Transformer-big) and decayed as the inverse square root of the number of updates after 16,000 updates. When training the Transformer model, a linearly growing learning rate is used during the first 16,000 iterations, starting with 0 until the base learning rate is reached.

Table 1 contains our results for WMT2017 training data with back-translation. We match re-

[1] Proprocessing was performed using scripts from Moses (Koehn et al., 2007).

[2] `https://github.com/rsennrich/subword-nmt`

[3] R2L training, scoring or decoding does not require data processing, right-to-left inversion is built into Marian.

[4] `https://github.com/mjpost/sacreBLEU`

sults from Sennrich et al. (2017) with our re-implementation of their models (Deep RNN) and outperform them with base and big Transformer versions. Differences between the best Deep RNN model and Transformer-big reach up to 2 BLEU points for the complete system. Ensembling is quite effective, right-to-left reranking seems to be moderately effective for Transformer models.

4 Taking advantage of Paracrawl

This year's shared task included a new, large but somewhat noisy parallel resource: Paracrawl. First experiments with shallow RNN models (chosen for fast experimentation) indicated that adding this data without a rigorous data filtering scheme would lead to catastrophic loss in quality (compare WMT+back-trans and Paracrawl-32M in Table 2). We therefore experiment with data selection and weighting.

4.1 Dual conditional cross-entropy filtering

The scoring method introduced in this section is our main contribution to the WMT2018 Shared Task on Parallel Corpus Filtering (Koehn et al., 2018), details are provided in our corresponding system submission (Junczys-Dowmunt, 2018).

For a sentence pair (x, y) we calculate a score:

$$\begin{aligned} &|H_A(y|x) - H_B(x|y)| \\ +&\frac{1}{2}\left(H_A(y|x) + H_B(x|y)\right) \end{aligned} \tag{1}$$

where A and B are translation models trained on the same data but in inverse directions, and $H_M(\cdot|\cdot)$ is the word-normalized conditional cross-entropy of the probability distribution $P_M(\cdot|\cdot)$ for a model M:

$$\begin{aligned} H_M(y|x) &= -\frac{1}{|y|}\log P_M(y|x) \\ &= -\frac{1}{|y|}\sum_{t=1}^{|y|}\log P_M(y_t|y_{<t}, x). \end{aligned}$$

The score (denoted as dual conditional cross-entropy) has two components with different functions: the absolute difference $|H_A(y|x) - H_B(x|y)|$ measures the agreement between the two conditional probability distributions, assuming that (word-normalized) translation probabilities of sentence pairs in both directions should be roughly equal. We want disagreement to be low, hence this value should be close to 0.

However, a translation pair that is judged to be equally improbable by both models will also have a low disagreement score. Therefore we weight the agreement score by the average word-normalized cross-entropy from both models. Improbable sentence pairs will have higher average cross-entropy values.

This score is also quite similar to the dual learning training criterion from He et al. (2016) and Hassan et al. (2018). The dual learning criterion is formulated in terms of joint probabilities, later decomposed into translation model and language model probabilities. In practice, the influence of the language models is strongly scaled down which results in a form more similar to our score.

While Moore and Lewis filtering requires an in-domain data set and a non-domain-specific data set to create helper models, we require a clean, relative high-quality parallel corpus to train the two dual translation models. We sample 1M sentences from WMT parallel data excluding Paracrawl and train Nematus-style translation models $W_{\text{de}\rightarrow\text{en}}$ and $W_{\text{en}\rightarrow\text{de}}$.

Formula (1) produces only positive values with 0 being the best possible score. We turn it into a partial score with values between 0 and 1 (1 being best) by negating and exponentiating, setting $A = W_{\text{de}\rightarrow\text{en}}$ and $B = W_{\text{en}\rightarrow\text{de}}$:

$$\begin{aligned} \text{adq}(x, y) = \exp(-(&|H_A(y|x) - H_B(x|y)| \\ &+ \frac{1}{2}\left(H_A(y|x) + H_B(x|y)\right))). \end{aligned}$$

We score the entire Paracrawl data with this score and keep the scores. We further assign a value of 1 to all original WMT parallel sentences. That way we have a score for every sentence.

4.2 Cross-entropy difference filtering

We treated cross-entropy filtering proposed by Moore and Lewis (2010) as another score. Cross-entropy filtering or Moore-Lewis filtering uses the quantity

$$H_I(x) - H_N(x) \tag{2}$$

where I is an in-domain model, N is a non-domain-specific model and H_M is the word-normalized cross-entropy of a probability distribution P_M defined by a model M:

$$\begin{aligned} H_M(x) &= -\frac{1}{|x|}\log P_M(x) \\ &= -\frac{1}{|x|}\sum_{t=1}^{|x|}\log P_M(x_t|x_{<t}). \end{aligned}$$

Sentences scored with this method and selected when their score is below a chosen threshold are likely to be more in-domain according to model I and less similar to data used to train N than sentences above that threshold.

We chose WMT German news data from the years 2015-2017 as our in-domain, clean language model data and sampled 1M sentences to train model $I = W_{\text{en}}$. We sampled 1M sentences from Paracrawl without any previously applied filtering to produce $N = P_{\text{de}}$.

To create a partial score for which the best sentence pairs produce a 1 and the worst at 0, we apply a number of transformations. First, we negate and exponentiate cross-entropy difference arriving at a quotient of perplexities of the target sentence y (x is ignored):

$$\begin{aligned}\text{dom}'(x,y) &= \exp(-(H_I(y) - H_N(y))) \\ &= \frac{\text{PP}_N(y)}{\text{PP}_I(y)}.\end{aligned}$$

This score has the nice intuitive interpretation of how many times sentence y is less perplexing to the in-domain model W_{de} than to the out-of-domain model P_{de}.

We further clip the maximum value of the score to 1 (the minimum value is already 0) as:

$$\text{dom}(x,y) = \max(\text{dom}'(x,y), 1). \quad (3)$$

This seems counterintuitive at first, but is done to avoid that a high monolingual in-domain score strongly overrides bilingual adequacy; we are fine with low in-domain scores penalizing sentence pairs. This is a precision-recall trade-off for adequacy and we prefer precision.

We score the entire parallel data, Paracrawl, back-translated data and previous WMT data with this score. Next we multiply the adequacy and domain-based score to obtain a single score for all parallel data and all Paracrawl data in particular.

4.3 Data selection

Based on the scores produced in the previous section, we sort the new Paracrawl data by decreasing scores from 1 to 0. Next we select the first N sentences from the sorted corpus, add it to WMT and back-translated data and train again a shallow RNN model. In our experiments it seems, that selecting the first 8M out of 32M sentences according to this score leads to the largest gains on WMT2016 test data. A loss of 2.5 BLEU on full WMT+Paracrawl data is turned into a gain of 1.4 BLEU on WMT with selected Paracrawl data (see +Paracrawl-8M in Table 2).

Data	2016
WMT+back-trans.	32.6
+Paracrawl-32M	30.1
+Paracrawl-2M	33.2
+Paracrawl-4M	33.5
+Paracrawl-8M	**34.0**
+Paracrawl-16M	31.9
+Paracrawl-24M	30.3
+Paracrawl-8M-weights	**34.2**
+Paracrawl-24M-weights	33.4

Table 2: Effects of data cleaning, filtering and weighting on BLEU. Evaluated with default shallow Nematus-style RNN model

4.4 Data weighting

We further experiment with sentence instance weighting, a feature available in Marian. Here we use the computed score for a sentence pair as a multiplier of the per-sentence cross-entropy cost during training. Sentences with high scores will contribute more to the training, sentence with low cost contribute less. Scores are however clipped at 1, so no score can contribute more than it would without weighting. As stated above, sentences from original WMT training data and from back-translation have an adequacy score of 1, so they are only weighted by their domain multiplier. Sentences from Paracrawl are weighted by a product of their adequacy and domain score. We see slight improvements for +Paracrawl-8M-weights over the unweighted version. It also seems that weighting can at least partially eliminate harmful effects from bad data. The 24M variant is far less damaging than the unweighted version. This seems worth to be explored in future work.

5 Final submission

We chose the +Paracrawl-8M-weights setting as our training setting for the Transformer-big configuration. Training and model parameters remain the same, we only add 8M Paracrawl sentences and sentence-level scores for all parallel sentences and retrain all models. In Table 3, we see that compared to Table 1 the Transformer-big model can

System	2016	2017	2018*
Transformer-big (x1)	38.6	31.3	46.5
+Ensemble (x4)	39.3	31.6	47.9
+R2L Reranking (x4)	39.3	31.7	48.0
+Transformer-LM	**39.6**	**31.9**	**48.3**

Table 3: Best model retrained on WMT and selected Paracrawl data. Sentences are weighted. Asterisk * marks post-submission evaluation.

take even more advantage of the filtered, selected and weighted data than the shallow models we used for development. We gain 1 to 2.5 BLEU points on the different test sets. Right-to-left re-ranking seems to matter less, however these models had not yet fully converge at time of submission.

5.1 Ensembling with a Transformer-style language model

We also experiment with shallow-fusion[5] or ensembling with a language model. We train a Transformer-style language model with Marian, following the architecture of the Transformer-big decoder without target-source attention blocks. We observed that this type of model has lower perplexity than LSMT models with similar numbers of parameters. We use 100M German monolingual sentences from 2016-2018 news data and train for two full epochs.

The resulting language model is ensembled with the left-to-right translation models at decoding time. We determine an optimal weight of 0.4 on a the newstest2016. The other models in the ensemble have a weight of 1. Since scores are summed it is a 4 to 0.4 ratio for translation models versus language model log probabilities. We see that the language model has a small, but consistently positive effect on all test sets of 0.2-0.3 BLEU.

6 Results

According to the automatically calculated BLEU scores on the WMT submission page, we achieve the highest BLEU score for English-German by a large margin over the next best system. We include the results for the 7 best systems in Table 4. The next best systems are quite tightly packed. We also rank highest among constrained systems based on human evaluation (Table 5).

[5] We do not like this term, in the end this is just ensembling.

System	BLEU
Microsoft-Marian	**48.3**
UCAM	46.6
NTT	46.5
KIT	46.3
MMT-PRODUCTION	46.2
UEDIN	44.4
JHU	43.4

Table 4: Automatic BLEU scores from submission page for 7 best submissions. There were 21 submissions in total.

Rank	Ave. %	Ave. z	System
2	**81.9**	**0.551**	**Microsoft-Marian**
	82.3	0.537	UCAM
	80.2	0.491	NTT
	79.3	0.454	KIT
8	76.7	0.377	JHU
	76.3	0.352	UEDIN
11	71.8	0.213	LMU-NMT
15	36.7	-0.966	RWTH-UNSUP
16	32.6	-1.122	LMU-UNSUP

Table 5: Human evaluation of constrained systems. Unconstrained systems have been omitted, see Bojar et al. (2018) for full list.

7 Conclusions

It seems strong state-of-the-art models and data hacking are winning combinations. Our data filtering method – developed first for this system – also proved very effective during the Parallel Corpora Filtering Task and we believe it had a large influence on our current result.

References

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. arXiv preprint arXiv:1607.06450.

Ondrej Bojar, Christian Buck, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno-Yepes, Philipp Koehn, and Julia Kreutzer, editors. 2017. Proc. of the 2nd Conference on Machine Translation, WMT 2017. 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, Brussels, Belgium. Association for Computational Linguistics.

Yarin Gal and Zoubin Ghahramani. 2016. A theoretically grounded application of dropout in recurrent neural networks. In Advances in neural information processing systems, pages 1019–1027.

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.

Di He, Yingce Xia, Tao Qin, Liwei Wang, Nenghai Yu, Tieyan Liu, and Wei-Ying Ma. 2016. Dual learning for machine translation. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, Advances in Neural Information Processing Systems 29, pages 820–828. Curran Associates, Inc.

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, Brussels, Belgium. 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.

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 ACL. The Association for Computer Linguistics.

Philipp Koehn, Huda Khayrallah, Kenneth Heafield, and Mikel Forcada. 2018. 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.

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.

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, volume 2, pages 157–163.

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, WMT 2017, pages 389–399.

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, 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.