NTT's Neural Machine Translation Systems for WMT 2018

Makoto Morishita, Jun Suzuki* and **Masaaki Nagata**
NTT Communication Science Laboratories, NTT Corporation, Japan
{morishita.makoto, nagata.masaaki}@lab.ntt.co.jp
jun.suzuki@ecei.tohoku.ac.jp

Abstract

This paper describes NTT's neural machine translation systems submitted to the WMT 2018 English-German and German-English news translation tasks. Our submission has three main components: the Transformer model, corpus cleaning, and right-to-left n-best re-ranking techniques. Through our experiments, we identified two keys for improving accuracy: filtering noisy training sentences and right-to-left re-ranking. We also found that the Transformer model requires more training data than the RNN-based model, and the RNN-based model sometimes achieves better accuracy than the Transformer model when the corpus is small.

1 Introduction

This paper describes NTT's submission to the WMT 2018 news translation task (Bojar et al., 2018). This year, we participated in English-to-German (En-De) and German-to-English (De-En) translation tasks. The starting point of our system is the Transformer model (Vaswani et al., 2017), which recently established better performance than conventional RNN-based models (Sutskever et al., 2014; Bahdanau et al., 2015; Luong et al., 2015). We incorporated a parallel corpus cleaning technique (Section 3.1) and a right-to-left n-best re-ranking technique (Section 3.4) and also used a synthetic corpus to exploit monolingual data. To maintain the quality of the synthetic corpus, we checked its back-translation BLEU scores and filtered out the noisy data with low scores (Section 3.2).

Through experiments, we evaluated how each feature affects accuracy (Section 4). Compared with the RNN-based system, we also identified when the Transformer model works effectively (Section 4.3.3).

*His current affiliation is Tohoku University.

2 Neural Machine Translation

Neural Machine Translation (NMT) has been making rapid progress in recent years. Sutskever et al. (2014) proposed the first NMT model that uses a simple RNN-based encoder-decoder network. Luong et al. (2015); Bahdanau et al. (2015) augmented this architecture with an attention mechanism, allowing the decoder to refer back to the encoder-side information at each time step. These conventional NMT models use RNNs as encoder and decoder to model sentence-level information. However, the RNN-based model uses previous states for predicting subsequent target words, which can cause a bottleneck in efficiency. Recently, Vaswani et al. (2017) proposed a model called Transformer, which completely relies on attention and feed-forward layers instead of RNN architecture. This model enables evaluation of a sentence in parallel by removing recurrence in the encoder/decoder, and we can train the model significantly faster than RNN-based models. It also established a new state-of-the-art performance in WMT 2014 translation tasks while shortening the training time by its GPU efficient architecture. In preliminary experiments, we also confirmed that the Transformer model tends to achieve better accuracy than RNN-based models, and thus we changed our base model for 2018 to the Transformer. For further details and formulation on the Transformer model, see Vaswani et al. (2017).

3 System Features

This year's submission includes the following features:

- Noisy data filtering for Common Crawl and ParaCrawl corpora (Section 3.1).
- Synthetic parallel data from the monolingual corpus (News Crawl 2017) with

Proceedings of the Third Conference on Machine Translation (WMT), Volume 2: Shared Task Papers, pages 461–466

https://doi.org/10.18653/v1/W18-64048

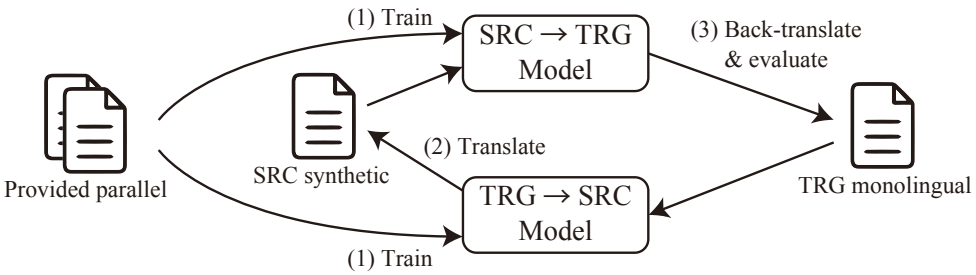

Figure 1: Overview of back-translation BLEU-based synthetic corpus filtering

back-translation BLEU-based filtering (Section 3.2).

- n-best re-ranking by a right-to-left translation model (Section 3.4).

From here, we discuss these features and experimentally verify each one.

3.1 Noisy Data Filtering

This year, ParaCrawl and Common Crawl corpora, which were created by crawling parallel websites, were provided for training. Since these web-based corpora are large but noisy, it seems essential to filter out noisy sentence pairs. Since the ParaCrawl corpus has already been cleaned by Zipporah (Xu and Koehn, 2017), we chose another method for further cleaning[1].

To clean the corpus, we selected the `qe-clean`[2] toolkit (Denkowski et al., 2012), which uses a language model to evaluate a sentences naturalness and a word alignment model to check whether the sentence pair has the same meaning. Both models are trained with clean data for scoring possibly noisy parallel sentence pairs and removes sentences with scores below a threshold. For more details, see Denkowski et al. (2012).

We used Europarl, News Commentary, and Rapid corpora as clean parallel data for training the word alignment model. We also used News Crawl 2017 as an additional monolingual corpus for language modeling. Since our target is news translation, using a news-related monolingual corpus is beneficial to train language models. We used `KenLM` (Heafield, 2011) and `fast_align` (Dyer et al., 2013, 2010) for language modeling and word alignment. To find the appropriate weights for each feature, we used newstest 2017 as a development set and fixed the threshold as one standard deviation.

3.2 Synthetic Corpus

One drawback of NMT is that it can only be trained with parallel data. Using synthetic corpora, which are pseudo-parallel corpora created by translating monolingual data with an existing NMT model, is one of the ways to make use of monolingual data (Sennrich et al., 2016a). We created a synthetic corpus by translating monolingual sentences with a target-to-source translation model and used it as additional parallel data.

In our case, we trained a baseline NMT model with a provided parallel corpora[3] and translated News Crawl 2017 to make a synthetic corpus.

3.3 Back-translation BLEU-based Filtering for Synthetic Corpus

A synthetic corpus might contain noise due to translation errors. Since these noisy sentences might deleteriously affect the training, we filtered them out.

In this work, we did back-translation BLEU-based synthetic corpus filtering (Imankulova et al., 2017). We hypothesize that synthetic sentence pairs can be correctly back-translated to the target language unless they contains translation errors. Based on this hypothesis, we found better synthetic sentence pairs by evaluating how the back-translated sentences resembled the original source sentences.

Figure 1 shows an overview of our synthetic corpus filtering process. First, we trained the NMT model with the provided parallel corpora and then translated the monolingual sentences in the target language to the source language by a target-to-

[1] Although the provided ParaCrawl corpus was already filtered by Zipporah (Xu and Koehn, 2017), a cursory glance suggested that it still contains many noisy sentence pairs.

[2] https://github.com/cmu-mtlab/qe-clean

[3] Europarl + News Commentary + Rapid + a filtered version of Common Crawl and ParaCrawl corpora

source translation model. After getting the translation, we back-translated it with the source-to-target model. Then we evaluated how well it restored the original sentences by sentence-level BLEU scores (Lin and Och, 2004), selected the high-scoring sentence pairs, and created a synthetic corpus whose size equals the naturally occurring parallel corpus.

3.4 Right-to-Left Re-ranking

Liu et al. (2016) pointed out that RNN-based sequence generation models lack reliability when decoding the end of the sentence. This is due to its autoregressive architecture that uses previous predictions as context information. If the model makes a mistake, this error acts as a context for additional predictions, often causing further errors.

To alleviate this problem, Liu et al. (2016) proposed a method that re-ranks an n-best hypothesis generated by the Left-to-Right (L2R) model, which generates a sentence from its beginning (left) to its end (right), by the Right-to-Left (R2L) model that generates a sentence in the opposite order. Their work mainly focuses on the problem of RNN-based models and the effect is unclear when applied to the Transformer model, which completely relies on attention and feed-forward layers. We assume this method also works with the Transformer model because it still has autoregressive architecture in its decoding phase.

We re-ranked the n-best hypothesis of the L2R model by the R2L model with the following formula:

$$P(\tilde{y}) = \arg\max_{y \in \boldsymbol{Y}} P(y|x;\theta_{L2R})P(y^r|x;\theta_{R2L}), \quad (1)$$

where $\boldsymbol{Y}$ is a set of n-best translations of source sentence x obtained by the L2R model, y^r is a reversed sentence of y, and θ_{L2R} and θ_{R2L} are the model parameters for the L2R and R2L models, respectively. In our experiments, we set $n = 10$.

4 Experiments

4.1 Data

As the first step of our data preparation, we applied the moses-tokenizer[4] and the truecaser[5] to all the datasets used in our experiments. Then we split the words into subwords by joint Byte-Pair-Encoding (BPE) (Sennrich et al., 2016b) with 32,000 merge operations. Finally, we discarded from the training data the sentence pairs that exceed 80 subwords either in the source or target sentences. As a development set, we used newstest 2017 (3004 sentences).

4.2 Translation model

Transformer We used the `tensor2tensor`[6] implementation to train the Transformer model. Our hyper-parameters are based on the previously introduced Transformer big setting (Vaswani et al., 2017), and we also referred Popel and Bojar (2018) for tuning hyper-parameters. We used six layers for both the encoder and the decoder. All the sub-layers and the embeddings layers output 1024 dimension vectors, and the inner-layer of the position-wise feed-forward layers has 4096 dimensions. For multi-head attention, we used 16 parallel attention layers. We use the same weights for the encoder/decoder embedding layers and the decoder output layer by three-way-weight-tying (Press and Wolf, 2017). As an optimizer, we used Adam (Kingma and Ba, 2015) with $\beta_1 = 0.9$ and $\beta_2 = 0.997$ and set dropout (Srivastava et al., 2014) with a probability of 0.1. We used a learning rate decaying method proposed by (Vaswani et al., 2017) with 16,000 warm-up steps and trained the model for 300,000 steps. Each mini-batch contained roughly 20,000 tokens. We saved a model every hour and averaged the last 16 model parameters for decoding. The training took about three days for both En-De and De-En with eight GTX 1080Ti GPUs. During decoding, we used a beam search with a size of ten and a length normalization technique (Wu et al., 2016) with $\alpha = 1.0$ and $\beta = 0.0$.

RNN-based In several experimental settings, we also trained an RNN-based attentional NMT model based on a previous work (Luong et al., 2015) for comparison[7]. We used a two-layer LSTM-based model and respectively set the embedding and hidden layer unit sizes to 512 and 1024. As an optimizer, we used SGD and set an initial learning rate to 1.0. We decayed the learn-

[4] https://github.com/moses-smt/mosesdecoder/blob/master/scripts/tokenizer/tokenizer.perl

[5] https://github.com/moses-smt/mosesdecoder/blob/master/scripts/recaser/truecase.perl

[6] https://github.com/tensorflow/tensor2tensor

[7] Implementation and settings are based on our submission to WAT shared-task (Morishita et al., 2017).

ing rate after 13 epochs by multiplying 0.7 per epoch and trained the model for 20 epochs. We clipped the gradient (Pascanu et al., 2013) if its norm exceeded 5.0. We set the dropout probability to 0.3. Each mini-batch contained about 128 sentences. The training took about 23 days for De-En and 31 days for En-De on a single GTX 1080Ti GPU. During decoding, we set the beam size to 20 and normalized the scores by dividing them by the sentence length.

4.3 Experimental Results and Discussions

Table 1 shows the provided and filtered corpus sizes for training. The Original Common Crawl and ParaCrawl corpora contain around 35.56M sentences. However, since most of the sentence pairs are noisy, we only retained the cleanest 4.01M sentences that were selected by the `qe-clean` toolkit. For the synthetic corpus, we chose the same size as the filtered parallel corpus based on the back-translation BLEU+1 scores.

Table 2 shows the evaluation results of our submission and baseline systems. Here, we report the case-sensitive BLEU scores (Papineni et al., 2002) evaluated by the provided automatic evaluation system[8]. In the following, unless specified, we mainly discuss the Transformer model results.

4.3.1 Effect of Corpus Filtering

We split the provided corpora into two parts: (1) Europarl, News Commentary and Rapid corpora as clean, and (2) Common Crawl and ParaCrawl corpora as noisy.

First, we just trained the model with cleaner corpora (Setting `(1)`) and added possibly noisy corpora (Setting `(2)`). The noisy parallel corpus seriously damaged the model for En-De, although there was a small gain for De-En. After filtering out the noisy part of the corpora (Setting `(3)`, it showed a large gain of +11.3 points for En-De and +4.8 points for De-En compared to the unfiltered setting. This suggests that clean, small training data tend to outperform large but noisy data. This large gain might also come from the effect of domain adaptation. We used news-related monolingual sentences to train the language model for corpus filtering, and thus our filtered sentences are related to a news domain, which is the same as our test set.

Then we added a synthetic corpus with and without filtering (Settings `(4)` and `(5)`). Although adding an unfiltered corpus resulted in certain gain, we identified an additional gain of +3.5 points for En-De by filtering out low-quality synthetic sentence pairs based on back-translation BLEU+1 scores.

Synthetic corpus filtering worked well, especially for En-De; but we did not see a large difference for De-En. To determine why, we estimated the quality of the synthetic corpus by checking the back-translation BLEU+1 scores. Table 3 shows the average back-translation BLEU+1 scores of the filtered/unfiltered synthetic corpus. These scores reflect the translation accuracy of the synthetic sentences. Before filtering, the average En-De score was lower than the average De-En score. From this result, we suspect that De-Ens unfiltered synthetic corpus is clean enough, resulting in no improvement from further filtering. After choosing high-scoring sentence pairs, the average scores exceed 80 for both language pairs, ensuring the quality of the synthetic corpus.

From our experiments, we confirmed that noisy parallel sentence pairs significantly damaged the model. For the best results, noisy sentences must be filtered out before training the model.

4.3.2 Effect of Right-to-Left Re-ranking

By re-ranking the n-best hypothesis by the R2L model, we saw a gain of 1.5 points for En-De and 0.5 points for De-En (Setting `(6)`). We submitted these results as our primary submission.

R2L n-best re-ranking works well with the RNN-based model, but we confirmed that it also works well with the Transformer model. We suppose both the Transformer and the RNN models lack the ability to decode the end of the sentence, but R2L model re-ranking can alleviate this problem.

4.3.3 Comparison of Transformer and RNN

For settings `(1)`, `(3)`, and `(5)`, we also trained the RNN-based NMT for comparison. We compared the Transformer and the RNN and found the latter achieved comparable or sometimes better results than the Transformer when trained with a small parallel corpus (Settings `(1)` and `(3)`). When the corpus size increased after adding a synthetic corpus, Transformer surpassed the RNN (Setting `(5)`). Our results suggest that Transformer gets stronger when the parallel corpus is enough large, but it might be worse than the

[8] http://matrix.statmt.org/

Corpus	Sentences
Europarl + News Commentary + Rapid	3.10M
Common Crawl + ParaCrawl	35.56M
Filtered version of Common Crawl + ParaCrawl	4.01M
Synthetic corpus (News Crawl 2017)	37.94M (En-De), 25.86M (De-En)
Filtered version of synthetic corpus (News Crawl 2017)	7.11M

Table 1: Number of sentences in datasets

		En-De			De-En		
	Settings	Sentences	Transformer	RNN	Sentences	Transformer	RNN
(1)	Europarl + News Commentary + Rapid	3.10M	32.5	30.4	3.10M	31.0	31.0
(2)	(1) + Unfiltered Common Crawl + ParaCrawl	38.66M	26.6	—	38.66M	32.7	—
(3)	(1) + Filtered Common Crawl + ParaCrawl	7.11M	37.9	39.6	7.11M	37.5	39.6
(4)	(3) + Unfiltered synthetic corpus	45.05M	41.5	—	32.97M	46.4	—
(5)	(3) + Filtered synthetic corpus	14.22M	45.0	39.8	14.22M	46.3	43.7
(6)	(5) + R2L re-ranking (submission)	14.22M	46.5	—	14.22M	46.8	—

Table 2: Cased BLEU scores of our submission and baseline systems

	En-De	De-En
Unfiltered	44.02	53.96
Filtered	80.12	80.81

Table 3: Average back-translation BLEU+1 scores of synthetic corpus

RNN-based models when the corpus size is small. One critical reason is that Transformer has many trainable parameters, complicating training with small training data. This result might change with smaller hyper-parameter settings (e.g., Transformer base setting), but we set aside this idea for future work.

5 Conclusion

In this paper, we described our submission to the WMT 2018 news translation task. Through experiments, we found that careful parallel corpus cleaning for the provided and synthetic corpora largely improved accuracy, and we confirmed that R2L re-ranking works well even with the Transformer model. Our comparison between the Transformer and RNN-based models suggests that the latter models might surpass the former when the training data are not enough large. This result sheds light on the importance of large, clean data for training the Transformer model.

Acknowledgments

We thank two anonymous reviewers for their insightful comments and suggestions.

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)*.

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 3rd Conference on Machine Translation (WMT)*.

Michael Denkowski, Greg Hanneman, and Alon Lavie. 2012. The cmu-avenue french-english translation system. In *Proceedings of the 7th Workshop on Statistical Machine Translation (WMT)*, pages 261–266.

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 (NAACL HLT)*, pages 644–648.

Chris Dyer, Adam Lopez, Juri Ganitkevitch, Jonathan Weese, Ferhan Ture, Phil Blunsom, Hendra Setiawan, Vladimir Eidelman, and Philip Resnik. 2010. cdec: A decoder, alignment, and learning framework for finite-state and context-free translation models. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 7–12.

Kenneth Heafield. 2011. KenLM: Faster and smaller language model queries. In *Proceedings of the 6th Workshop on Statistical Machine Translation (WMT)*, pages 187–197.

Aizhan Imankulova, Takayuki Sato, and Mamoru Komachi. 2017. Improving low-resource neural machine translation with filtered pseudo-parallel cor-

pus. In *Proceedings of the 4th Workshop on Asian Translation (WAT)*, pages 70–78.

Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*.

Chin-Yew Lin and Franz Josef Och. 2004. Orange: a method for evaluating automatic evaluation metrics for machine translation. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING)*, pages 501–507.

Lemao Liu, Andrew Finch, Masao Utiyama, and Eiichiro Sumita. 2016. Agreement on target-bidirectional lstms for sequence-to-sequence learning. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI)*, pages 2630–2637.

Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1412–1421.

Makoto Morishita, Jun Suzuki, and Masaaki Nagata. 2017. NTT neural machine translation systems at WAT 2017. In *Proceedings of the 4th Workshop on Asian Translation (WAT)*, pages 89–94.

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 (ACL)*, pages 311–318.

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 (ICML)*, volume 28, pages 1310–1318.

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

Ofir Press and Lior Wolf. 2017. Using the output embedding to improve language models. In *Proceedings of the 15th European Chapter of the Association for Computational Linguistics (EACL)*, pages 157–163.

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)*, 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 (ACL)*, pages 1715–1725.

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*, 15:1929–1958.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of the 28th Annual Conference on Neural Information Processing Systems (NIPS)*, 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 *Proceedings of the 31st Annual Conference on Neural Information Processing Systems (NIPS)*, pages 6000–6010.

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 preprint arXiv:1609.08144*.

Hainan Xu and Philipp Koehn. 2017. Zipporah: a fast and scalable data cleaning system for noisy web-crawled parallel corpora. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2945–2950.