AISTAI Neural Machine Translation Systems for WAT 2019

Wei Yang
Artificial Intelligence Research Center,
National Institute of Advanced Industrial
Science and Technology (AIST),
Tsukuba 305-8560, Japan
wei.yang@aist.go.jp

Jun Ogata
Artificial Intelligence Research Center,
National Institute of Advanced Industrial
Science and Technology (AIST),
Tsukuba 305-8560, Japan
jun.ogata@aist.go.jp

Abstract

In this paper, we describe our Neural Machine Translation (NMT) systems for the WAT 2019 translation tasks we focus on. This year we participate in scientific paper tasks and focus on the language pair between English and Japanese. We use Transformer model through our work in this paper to explore and experience the powerful of the Transformer architecture relying on self-attention mechanism. We use different NMT toolkit/library as the implementation of training the Transformer model. For word segmentation, we use different subword segmentation strategies while using different toolkit/library. We not only give the translation accuracy obtained based on absolute position encodings that introduced in the Transformer model, but also report the the improvements in translation accuracy while replacing absolute position encodings with relative position representations. We also ensemble several independent trained Transformer models to further improve the translation accuracy.

1 Introduction

Machine translation (MT) is a specific task of natural language processing (NLP). It is used to automatically translate speech or text from one natural language to another natural language using translation system. In neural machine translation (NMT), different from statistical machine translation (SMT), deep learning is done using neural network technology. In the last five years, statistical machine translation is gradually fading out in favor of neural machine translation. Google translate supports over 100 languages. In November 2016, Google has switched to a neural machine translation engine for 8 languages firstly between English (to and from) and Chinese, French, German, Japanese, Korean, Portuguese, Spanish and Turkish[1]. By July 2017 all languages support translation to and from English by GNMT (Wu et al., 2016).

In our work, we focus on the NMT system constructed based on the Transformer model (Vaswani et al., 2017). The Transformer model use a different neural network architecture with self-attention mechanism. Different from sequence-aligned recurrent neural networks (RNNS) or convolution, the Transformer model computes representations of a sequence by considering different positions of the sequence relying on self-attention mechanism. All NMT experiments in our work are performed by using this state-of-the-art new network architecture.

"Tensor2Tensor[2]" (Vaswani et al., 2018) is a library for deep learning models that is widely used for NMT recently and includes the implementation of the Transformer model (Vaswani et al., 2017). It is used to train Transformer models and obtain the state-of-the-art translation accuracy for WMT[3] shared tasks: English-to-German and English-to-French on newstest2014 tests.

An open source for NMT and neural sequence learning, "OpenNMT[4]" has been released by the Harvard NLP group (Klein et al., 2017), and it provides implementations in 2 popular deep learning frameworks: PyTorch[5] (OpenNMT-py[6]) and TensorFlow[7] (Abadi et al., 2016) (OpenNMT-tf[8]). It has been extended to support many additional models and features including the Transformer

[1] https://en.wikipedia.org/wiki/Google_Neural_Machine_Translation
[2] https://github.com/tensorflow/tensor2tensor
[3] http://www.statmt.org/wmt19/
[4] http://opennmt.net/
[5] https://github.com/pytorch/pytorch
[6] https://github.com/OpenNMT/OpenNMT-py
[7] https://www.tensorflow.org/
[8] https://github.com/OpenNMT/OpenNMT-tf

Proceedings of the 6th Workshop on Asian Translation, pages 159–164
Hong Kong, China, November 4, 2019.

model, each implementation has its own set of features[9]. According to the needs in our experiments such as "Relative position representations" in model configuration and "Ensemble" in decoding, we choose to use "OpenNMT-py". We give the description of these two terms ("Relative position representations" and "Ensemble") in the following section.

In the last five years, two parallel corpora were released in the domain of scientific papers and patents. They are provided to promote machine translation research, including the condition of participating in the open evaluation campaign Workshop on Asian Translation (WAT) (Nakazawa et al., 2018, 2019). The first parallel corpus which provided for WAT from 2014 is the Asian Scientific Paper Excerpt Corpus (ASPEC) (Nakazawa et al., 2016). It contains 680,000 Japanese–Chinese parallel sentences used for training and approximately 3,000,000 English–Japanese training data extracted from scientific papers. The second parallel corpus provided for WAT from 2015 is the JPO corpus, created jointly, based on an agreement between the Japan Patent Office (JPO) and NICT. In our work, we propose to train several NMT systems between English and Japanese by leveraging a part of ASPEC-JE training data and several techniques. We also compare the translation accuracy between these systems so as to significantly improve the performance of NMT in scientific and technical domain.

Section 2 further introduces the background of our NMT systems and some related work. In section 3, we present the experiments and report the results by adding each technique or combine several techniques which are described in Sec. 2. Section 4 gives the conclusion and some future work.

2 Translation systems

2.1 Subword segmentation

Word segmentation (tokenization), i.e., breaking sentences down into individual words (tokens), is normally treated as the first step of preprocessing for natural language processing (NLP). For English and Japanese, in our experiments, we use scripts in Moses (Koehn et al., 2007) and Juman[10] as the basic segmentation toolkits for word segmentation (tokenization), and then we perform subword segmentation to further segment words for preparing the final experimental data. Because some previous work has used subwords as a way for addressing unseen word or rare word problems in machine translation (Sennrich et al., 2016b), reducing model size (Wu et al., 2016), or as one of the performance of training a independent translation model (Denkowski and Neubig, 2017) so as to obtain a stronger translation system. The investigation in the relationship between the choice of using "word-based" or "subword-based" segmentation strategy and the improvement of machine translation (MT) is conducted. It is conclude that the "subword-based" segmented data and the byte pair encoding (BPE) compression algorithm (Gage, 1994) that the segmenter relied on is effective and affects MT performance (Sennrich et al., 2016b).

In our experiments, we also examine the impact of "subword-based" strategy in technical and scientific domain. Because as we known, there exist a large amount technical words/terms in scientific paper and this may lead to some rare word or unknown word problems in MT. Thus, "subword-based" segmentation strategy can be very helpful for the translation of these words. We use BPE in "OpenNMT-py" and "wordpieces" (Schuster and Nakajima, 2012; Wu et al., 2016) in "Tensor2Tensor".

2.2 Relative position representations

Due to the Transformer is a different neural network architecture compare with recurrent neural network (RNN) and convolutional neural network (CNN), adding position information to its inputs is necessary and crucial important in model construction. In (Shaw et al., 2018), it is demonstrated that the way of introducing relative position representations, and instead of using absolute position encodings, using relative position representations in self-attention mechanism of the Transformer yields improvements of 1.3 BLEU and 0.3 BLEU respectively on the WMT 2014 English-to-German and English-to-French translation tasks.

In our experiments, we shall use a similar idea on ASPEC-JE tasks. In other words, we try to use absolute position encodings or relative position representations independently for training our Transformer models, but for WAT 2019 ASPEC translation tasks: English-to-Japanese and Japanese-to-English. There exist several previous

[9] http://opennmt.net/features/

[10] http://nlp.ist.i.kyoto-u.ac.jp/index.php?JUMAN

work for WAT using this method for improving translation accuracy on scientific paper tasks such as (Li et al., 2018). Different from their experiments, we use different size of the training data and development data at least and obtain better results in English-to-Japanese translation sub-task.

2.3 Ensemble technique

Some previous work shows improvements in BLEU scores for model ensembles (Sutskever et al., 2014; Sennrich et al., 2016a). The basic idea of ensemble technique is that training and decoding with multiple translation models. We propose to follow the idea given in (Denkowski and Neubig, 2017), combine several techniques and imply them in ASPEC-JE shared tasks. Thus, the final technique we explore is the ensemble of multiple independently trained, averaged translation models in prediction of the test set.

2.4 Evaluation

The main metric used in our experiments for automatically evaluating the translation outputs is BLEU (Papineni et al., 2002) method. All evaluation BLEU scores for translation results given in this paper are evaluated by WAT 2019 automatic evaluation system[11].

3 Experiments

We train and evaluate our model on the WAT 2019 scientific paper tasks, using the ASPEC-JE dataset consisting of approximately 3,000,000 lines of sentence pairs. It worth noticing that the data contained in the ASPEC-JE training corpus are not all perfect aligned. Thus, we use the first 1,500,000 pairs of sentences with higher similarity scores which are calculated using the method given in (Utiyama and Isahara, 2007) as our training data (Train). We do not do any filtering for these sentences by length in words/tokens. All development data (Dev) and test data (Test) sets are used in performing experiments. Statistics on our experimental data sets are given in Table 1.

First of all, we use the "Tensor2Tensor" library for training and evaluating the Transformer models. We train translation models using "transformer (big)" hyperparameter setting. For the two experiments (English-to-Japanese and Japanese-to-English) 32k "wordpieces" are broken from

	ASPEC-JE	English	Japanese
Train	sentences (lines)	1,500,000	1,500,000
	length in words (avg. ± std.dev.)	26.01 ± 11.76	28.20 ± 12.32
Dev	sentences (lines)	3,574	3,574
	length in words (avg. ± std.dev.)	24.65 ± 11.50	26.72 ± 11.87
Test	sentences (lines)	1,812	1,812
	length in words (avg. ± std.dev.)	24.49 ± 11.28	26.32 ± 11.50

Table 1: Statistics on our experimental data sets (after tokenizing and lowercasing). Here, 'avg ± std.dev.' gives the average length of the sentences in words.

words. We train for 300,000 steps on 4 GPUs, it costs only about 27 hours for each experiment. During training, we save checkpoints every 1,000 steps. We average the last 8, 10 and 20 checkpoints for decoding the test set and report the best one. For evaluation, we use beam search with a beam size of 4 and length penalty α=0.6 (Wu et al., 2016). On the English-to-Japanese and Japanese-to-English subtasks, our "Transformer (big)" models achieve BLEU scores of 42.92 (average the last 20 models) and 29.01 (average the last 10 model) respectively. Compare with the result for English-to-Japanese sub-task (BLEU=42.87) given in WAT official evaluation[12], we obtained the similar result (BLEU=42.92) to a certain extend. But we also give the translation result for Japanese-to-English direction (BLEU=29.01) which is not given in WAT official evaluation[13].

Except this two experiments, the following, a series of experiments are all performed using "OpenNMT-py" as shown in Table 2, thus, we do not mention it every time.

We then measure the effect of BPE by training "word-based" and "subword-based" systems using "OpenNMT-py" with the same 1,500,000 training data (without any cleaning). All options and parameters used in "OpenNMT-py" are set refer to "transformer (big)" hyperparameter setting in "Tensor2Tensor". These two systems ("word-based" and "subword-based") are considered as two baselines ("weak baseline" and "stronger baseline") of the following experiments. The only difference from the "weak baseline", the "stronger baseline" use BPE segmentation with 32k vocabulary both for English and Japanese. Scores for

[11] http://lotus.kuee.kyoto-u.ac.jp/WAT/evaluation/index.html

[12] http://lotus.kuee.kyoto-u.ac.jp/WAT/evaluation/list.php?t=1&o=1

[13] http://lotus.kuee.kyoto-u.ac.jp/WAT/evaluation/list.php?t=2&o=4

Toolkit/library	System opts.	En-Ja	Ja-En
Tensor2Tensor	wordpieces + transformer (big)	42.92	29.01
OpenNMT-py	word-based + dropout=0.3 + warm-up=8,000 (weak baseline)	38.73	26.70
OpenNMT-py (1)	BPE + dropout=0.3 + warm-up=8,000 (stronger baseline)	41.91	28.92
OpenNMT-py (2)	BPE + Relative Position + dropout=0.1 + warm-up=8,000	42.06	28.64
OpenNMT-py (3)	BPE + Relative Position + dropout=0.3 + warm-up=8,000	42.83	28.86
OpenNMT-py (4)	BPE + Relative Position + dropout=0.3 + warm-up=16,000	42.63	28.32
OpenNMT-py	Ensemble (1) and (3)	43.15	29.36
OpenNMT-py	Ensemble (1), (2) and (3)	**43.76**	29.54
OpenNMT-py	Ensemble (1), (2) , (3) and (4)	43.60	**29.71**

Table 2: BLEU scores for ASPEC-JE test set using the Transformer (model) based NMT.

single models are averaged after training step. For using "OpenNMT-py", we average 4-6 saved models (models are saved per 10,000 steps, each experiments are trained for 160,000 steps.) with higher validation accuracy. Because we found that this may lead to better translation accuracy. As the results, compare with our "weak baseline", we improved translation accuracy by 3.2 and 2.2 BLEU points for both directions (English↔Japanese) by directly applying BPE subword segmentation strategy for English and Japanese.

But we found that the big transformer model (Transformer (big) in Table 2) training by "Tensor2Tensor" outperforms the reported models training by "OpenNMT-py" (the fourth line in Table 2), especially for English-to-Japanese (42.92 vs. 41.91).

After that, we compare our models using absolute position (sinusoidal position encodings) to using relative position representation instead. For English-to-Japanese, this approach improved nearly 1 BLEU points (41.91→42.83) compare with the "stronger baseline". In this experiment, for Japanese-to-English, there is no improvement in BLEU even slightly decreased compare with the "stronger baseline" system (28.92 →28.86).

For English-to-Japanese, "dropout" setting, we begin with 0.3 and then change it with 0.1, for "warm-up" setting, we begin with 8,000 and then change it with 16,000, we apply the same changing for Japanese-to-English. In other words, we do not touch any other settings and perform another two groups of experiments with the same data for both directions by only modifying the "dropout" value ("dropout=0.3" ⇒ "dropout=0.1" ("warm-up=8,000")) and "warm-up" value ("warm-up=8,000" ⇒ "warm-up=16,000" ("dropout=0.3"). As the results in both directions, "BPE + Relative position + dropout=0.3 + warm-up=8,000" allow us to obtain better BLEU scores compare with our "stronger baseline" system, especially for English-to-Japanese. In our experiments, translation accuracy are negatively affected by whatever changing "dropout" from 0.3 to 0.1 or "warm-up" form 8,000 to 16,000.

However, ensemble these models allow us to obtain our best BLEU scores for English-to-Japanese and Japanese-to-English sub-tasks. Again, all of these models are independent trained, averaged models. In Table 2, we show that combing these techniques can lead to significant improvements of over 1.85 BLEU score for English-to-Japanese by ensembling 3 independent models, 0.79 BLEU points for Japanese-to-English translation by ensembling 4 independent models. Boldface indicates the best BLEU scores over the two baseline systems. If we compare our final results with the "weak baseline" systems, combine using the Transformer model and several introduced efficient techniques, we obtained even more significant improvements by 5 and 3 BLEU points for English-to-Japanese and Japanese-to-English respectively in scientific domain.

As we mentioned in Sec. 2.4, all evaluation BLEU scores given in Table 2 are evaluated by WAT 2019 official automatic evaluation system[14]. We published the best two BLEU scores (43.76 and 29.71 for English-to-Japanese and Japanese-to-English) obtained by ensembling models using "OpenNMT-py", as well as another two BLEU scores (42.92 and 29.01 for English-to-Japanese and Japanese-to-English) obtained by "Tensor2Tensor" (transformer (big)).

[14] http://lotus.kuee.kyoto-u.ac.jp/WAT/evaluation/index.html

The translation result (BLEU=42.64)[15] submitted to WAT 2019 (Nakazawa et al., 2019) for both automatic evaluation and human evaluation obtained by an English-to-Japanese translation system using "Tensor2Tensor" (transformer (big)). We do not mention that system too much because it is only a test and very first system in our experiments with "transformer (big)" setting except the "train_steps" is only 131,000 (not 300,000 which allowed us to obtain BLEU=42.92).

4 Conclusion

The main focus of this paper is to exploit ASPEC-JE linguistic resources in technical and scientific domain and some existing technical methods to improve the translation accuracy between English and Japanese. In our experiments, we improved translation accuracy for WAT 2019 scientific paper tasks by using subword segmentation strategy, relative position representations and ensemble techniques, and tried to use the training data as small as possible without the use of any additional lexicon or additional corpus. We combined and applied several approaches and further improved the translation accuracy of English-to-Japanese and Japanese-to-English NMT by 1.85 and 0.79 BLEU scores compare with our "stronger baseline" systems, at the same time, we obtained 5 and 3 BLEU point improvements compare with our "weak baseline" systems. We found that we may obtain better translation accuracy by ensembling several independent models, even these models do not work very well independently. In future work, we propose to give in-depth analysis of where improvements obtained for translation results and give some statistics of them.

References

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2016. Tensorflow: A system for large-scale machine learning. In *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation*, OSDI'16, pages 265–283.

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.

Philip Gage. 1994. A new algorithm for data compression. *C Users J.*, 12(2):23–28.

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.

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.

Yihan Li, Boyan Liu, Yixuan Tong, Shanshan Jiang, and Bin Dong. 2018. SRCB neural machine translation systems in 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.

Toshiaki Nakazawa, Chenchen Ding, Raj Dabre, Hideya Mino, Isao Goto, Win Pa Pa, Nobushige Doi, Yusuke Oda, Anoop Kunchukuttan, Shantipriya Parida, Ondřej Bojar, and Sadao Kurohashi. 2019. Overview of the 6th workshop on Asian translation. In *Proceedings of the 6th Workshop on Asian Translation*, Hong Kong. Association for Computational Linguistics.

Toshiaki Nakazawa, Katsuhito Sudoh, Shohei Higashiyama, Chenchen Ding, Raj Dabre, Hideya Mino, Isao Goto, Win Pa Pa, Anoop Kunchukuttan, and Sadao Kurohashi. 2018. Overview of the 5th workshop on Asian translation. 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.

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 Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 2204–2208, Portorož, Slovenia.

Mike Schuster and Kaisuke Nakajima. 2012. Japanese and Korean voice search. In *International Conference on Acoustics, Speech and Signal Processing, IEEE (2012)*, pages 5149–5152.

[15] http://lotus.kuee.kyoto-u.ac.jp/WAT/evaluation/list.php?t=1&o=1

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.

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.

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.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. pages 3104–3112.

Masao Utiyama and Hitoshi Isahara. 2007. A Japanese-English patent parallel corpus. In *In proceedings of the Machine Translation Summit XI*.

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.

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 *NIPS'17 Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6000–6010, Long Beach, CA, USA.

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, Lukasz 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, Gregory S. 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.