NICT Self-Training Approach to Neural Machine Translation at NMT-2018

Kenji Imamura and **Eiichiro Sumita**
National Institute of Information and Communications Technology
3-5 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0289, Japan
{kenji.imamura,eiichiro.sumita}@nict.go.jp

Abstract

This paper describes the NICT neural machine translation system submitted at the NMT-2018 shared task. A characteristic of our approach is the introduction of self-training. Since our self-training does not change the model structure, it does not influence the efficiency of translation, such as the translation speed.

The experimental results showed that the translation quality improved not only in the sequence-to-sequence (seq-to-seq) models but also in the transformer models.

1 Introduction

In this study, we introduce the NICT neural translation system at the Second Workshop on Neural Machine Translation and Generation (NMT-2018) (Birch et al., 2018). A characteristic of the system is that translation qualities are improved by introducing self-training, using open-source neural translation systems and defined training data.

The self-training method discussed herein is based on the methods proposed by Sennrich et al. (2016a) and Imamura et al. (2018), and they are applied to a self training strategy. It extends only the source side of the training data to increase variety. The merit of the proposed self-training strategy is that it does not influence the efficiency of the translation, such as the translation speed, because it does not change the model structure. (However, the training time increases due to an increase in the training data size.)

The proposed approach can be applied to any translation method. However, we want to confirm on which model our approach is practically effective. This paper verifies the effect of our self-training method in the following two translation models:

- Sequence-to-sequence (seq-to-seq) models (Sutskever et al., 2014; Bahdanau et al., 2014) based on recurrent neural networks (RNNs). Herein, we use OpenNMT (Klein et al., 2017) as an implementation of the seq-to-seq model.
- The transformer model proposed by Vaswani et al. (2017). We used Marian NMT (Junczys-Dowmunt et al., 2018) as an implementation of the transformer model.

The remainder of this paper is organized as follows. Section 2 describes the proposed approach. Section 3 describes the details of our system. Section 4 explains the results of experiments, and Section 5 concludes the paper.

2 Self-training Approach

2.1 Basic Flow

The self-training approach in this study is based on a method proposed by Imamura et al. (2018). Their method extends the method proposed by Sennrich et al. (2016a) that a target monolingual corpus is translated back into source sentences and generates a synthetic parallel corpus. Then, the forward translation model is trained using the original and synthetic parallel corpora. The synthetic parallel corpus contains multiple source sentences of a target sentence to enhance the encoder and attention. The diversity of the synthetic source sentences is important in this study. Imamura et al. (2018) confirmed that the translation quality improved when synthetic source sentences were generated by sampling, rather than when they were generated by n-best translation.

Although Imamura et al. (2018) assumed the usage of monolingual corpora, it can be modified to a self-training form by assuming the target side of parallel corpora as monolingual corpora. In fact,

Proceedings of the 2nd Workshop on Neural Machine Translation and Generation, pages 110–115
Melbourne, Australia, July 20, 2018. ©2018 Association for Computational Linguistics

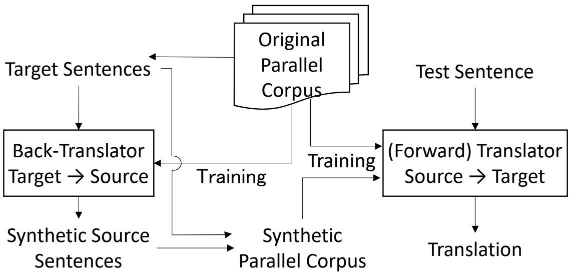


Figure 1: Flow of Self-training

they proposed such self-training strategy and confirmed the effect on their own corpus.

Figure 1 shows the flow of self-training. The procedure is summarized as follows.

1. First, train the back-translator that translates the target language into the source using original parallel corpus.

2. Extract the target side of the original parallel corpus, and translate it into the source language (synthetic source sentences) using the above back-translator. During back-translation, it not only generates one sentence but also generates multiple source sentences per target sentence using a sampling method.

3. Construct the synthetic parallel corpus making pairs of the synthetic source sentences and their original target sentences. If we define the number of synthetic source sentences per target sentence as N, the size of the synthetic parallel corpus becomes N-times larger than the original parallel corpus.

4. Train the forward translator, which translates the source to the target, using a mixture of the original and synthetic parallel corpora.

In this study, we modify the method proposed by Imamura et al. (2018) to improve the efficiency of the training while maintaining the diversity of the source sentences.

2.2 Diversity Control

Imamura et al. (2018) generates synthetic source sentences by sampling. The sampling is based on the posterior probability of an output word as follows.

$$y_t \sim \Pr(y|\boldsymbol{y}_{<t}, \boldsymbol{x}), \tag{1}$$

where y_t, $\boldsymbol{y}_{<t}$, and $\boldsymbol{x}$ denote the output word sequence at time t, history of the output words, and input word sequence, respectively.

To control the diversity of generated sentences, the synthetic source generation in this paper introduces an inverse temperature parameter $1/\tau$ into the softmax function.

$$y_t \sim \frac{\Pr(y|\boldsymbol{y}_{<t}, \boldsymbol{x})^{1/\tau}}{\sum_{y'} \Pr(y'|\boldsymbol{y}_{<t}, \boldsymbol{x})^{1/\tau}} \tag{2}$$

If we set the inverse temperature parameter $1/\tau$ greater than 1.0, high probability words become preferable, and if we set it to infinity, the sampling becomes identical to the argmax operation. On the contrary, if we set it less than 1.0, diverse words will be selected, and the distribution becomes uniform if we set it zero.

2.3 Dynamic Generation

A problem in the research proposed by Imamura et al. (2018) is that the training time increases $(N+1)$-times with an increase in the training data size. To alleviate this problem, we introduce dynamic generation that uses different synthetic parallel sets for each epoch (Kudo, 2018). Specifically, a synthetic parallel sentence set, which contains one synthetic source sentence per target sentence, is used for an epoch of the training. By changing the synthetic parallel sentence set for each epoch, we expect a similar effect to using multiple source sentences in the training.

For implementation, we do not embed the dynamic generation in the training program but perform it offline. Multiple synthetic source sentences were generated in advance, whose number N is 20 this time, and N synthetic parallel sets are constructed. During training, a synthetic set is selected for each epoch using round-robin scheduling, and learns the model using the synthetic set and the original corpus. We can use the same learning rates because the sizes of the original and synthetic sets are the same. Using the dynamic generation, the size of the training data is restricted to double of the original parallel corpus. [1]

The training procedure is summarized as follows.

1. First, train the back-translator using the original parallel corpus.

2. Translate the target side of the original corpus into N synthetic source sentences per target sentence using the above back-translator.

[1] Although the size is restricted double, the training time takes more than double because of late convergence.

Note that the sampling method described in Section 2.2 is used for the generation.

3. Make N sets of the synthetic parallel sentences by pairing the synthetic source sentences generated in Step 2 and the target side of the original parallel corpus.

4. Train the forward translator. In each epoch, select one synthetic parallel set, and train the model using the mixture of the synthetic and original parallel sets.

3 Applied Systems

In this paper, we apply the proposed self-training approach to two translator types; the seq-to-seq model (Sutskever et al., 2014; Bahdanau et al., 2014) implemented by OpenNMT (LUA version) (Klein et al., 2017) and the transformer model (Vaswani et al., 2017) implemented by Marian NMT (Junczys-Dowmunt et al., 2018). Table 1 summerizes the system description.

3.1 Back-Translator

The back-translator used herein is OpenNMT, which employs an RNN-based seq-to-seq model.

The training corpus for the back-translation is preprocessed using the byte-pair encoding (BPE) (Sennrich et al., 2016b). For each language, 16K subword types were independently computed. The model was optimized using the stochastic gradient descent (SGD) whose learning rate was 1.0.

For the back-translation, we modified OpenNMT to generate synthetic source sentences by sampling. This time, we generated three types of synthetic source sentences changing the inverse temperature parameter $1/\tau$ to 1.0, 1.2, and 1.5.

3.2 Forward Translator 1: Transformer Model

The first forward translator is Marian NMT, which is based on the transformer model. We used this system for the submission. The settings were almost identical to the base model of Vaswani et al. (2017). [2] The vocabulary sets were equal to those of the back-translator for the original parallel corpus. For the synthetic source sentences, we directly used subword sequences output from the back-translator.

Marian NMT performs the length normalization using the following equation.

$$ll_{\text{norm}}(\boldsymbol{y}|\boldsymbol{x}) = \frac{\sum_t \log \Pr(y_t|\boldsymbol{y}_{<t}, \boldsymbol{x})}{T^{WP}}, \quad (3)$$

where $ll_{\text{norm}}(\boldsymbol{y}|\boldsymbol{x})$, WP, and T denote the log-likelihood normalized by the output length, word penalty, and number of output words, respectively. If we set the word penalty greater than 0.0, long hypotheses are preferred. The setting of the word penalty will be further discussed in Section 4.1.

3.3 Forward Translator 2: Seq-to-Seq Model

The other forward translator used herein is OpenNMT based on the seq-to-seq model. The settings were almost the same as the back-translator. SGD was used for the optimization, but the learning rate was set to 0.5 because all target sentences appear twice in an epoch.

At the translation, we translated the source sentence into 10-best, and the best hypothesis was selected using the length reranking based on the following equation (Oda et al., 2017).

$$ll_{\text{bias}}(\boldsymbol{y}|\boldsymbol{x}) = \sum_t \log \Pr(y_t|\boldsymbol{y}_{<t}, \boldsymbol{x}) + WP \cdot T, \quad (4)$$

where $ll_{\text{bias}}(\boldsymbol{y}|\boldsymbol{x})$ denotes the log-likelihood biased by output length. Although this formula differs from Equation 3, there is an equivalent effect that long hypotheses are preferred if the word penalty WP is set to a positive value.

4 Experiments

In this section, we describe our results for the NMT-2018 shared task in English-German translation. Note that the shared task uses the WMT-2014 data set preprocessed by Stanford NLP Group.

In our experiments, we add two baselines. One is the model trained from the original parallel corpus only. Another is the model trained using the synthetic corpus, which contains 1-best generation and did not use the dynamic generation, with the original corpus. For the inverse temperature parameter $1/\tau$, we tested 1.0, 1.2, and 1.5. This is because the translation quality was better when the diversity was slightly inhibited in our preliminary experiments. Note that the submitted system was Marian NMT (the transformer model) trained using $1/\tau = 1.0$ synthetic corpus.

[2] We referred the settings described at https://github.com/marian-nmt/marian-examples/tree/master/transformer.

	Marian NMT	OpenNMT
Preprocessing	BPE 16K (independent)	BPE 16K (independent)
Model	Transformer	Seq-to-Seq
Word Embedding	512 units	500 units
Encoder	6-layer ($d_{model} = 512$, $d_{ff} = 2048$)	2-layer Bi-LSTM (500 + 500 units)
Decoder	6-layer ($d_{model} = 512$, $d_{ff} = 2048$)	2-layer LSTM (1,000 units)
Training	Adam (early stopping by cross-entropy)	SGD (14 + 6 epochs)
Learning Rate	0.0003	Back-translator: 1.0 Forward Translator: 0.5
Dropout	$d_{drop} = 0.1$	$d_{drop} = 0.3$
Maximum Length	100	80
Mini-batch Size	64	64
Translation	Beam Width: 6	Beam Width: 10

Program Arguments (for Training)	`--type transformer`
	`--max-length 100`
	`--mini-batch-fit --maxi-batch 1000`
	`--early-stopping 10`
	`--valid-freq 5000`
	`--valid-metrics cross-entropy perplexity translation`
	`--beam-size 6`
	`--enc-depth 6 --dec-depth 6`
	`--transformer-heads 8`
	`--transformer-postprocess-emb d`
	`--transformer-postprocess dan`
	`--transformer-dropout 0.1`
	`--label-smoothing 0.1`
	`--learn-rate 0.0003 --lr-warmup 16000`
	`--lr-decay-inv-sqrt 16000 --lr-report`
	`--optimizer-params 0.9 0.98 1e-09`
	`--clip-norm 5`
	`--sync-sgd --seed 1111`
	`--exponential-smoothing`
		`-brnn -brnn_merge concat`
		`-rnn_size 1000`
		`-end_epoch 20`
		`-start_decay_at 14`
		`-param_init 0.06`
		`-learning_rate 0.5`

Table 1: Summary of our Systems

4.1 Word Penalty / Length Ratio

The BLEU score significantly changes due to the translation length (Morishita et al., 2017). For instance, Figure 2 shows BLEU scores of our submitted system (a) when the word penalty was changed from 0.0 to 2.0 and (b) on various length ratios (LRs), which indicate the ratios of the number of words of the system outputs to the reference translations (sys/ref).

As shown in Figure 2 (a), the BLEU scores change over 0.5 when we change the word penalty. The penalties of the peaks are different among the development/test sets. The BLEU score peaks were at WP = 1.2, 0.2, and 0.5 in the `newstest2013`, `newstest2014`, and `newstest2015` sets, respectively. Therefore, the BLEU scores significantly depend on the word penalty. However, as shown in Figure 2 (b), we can see that the peaks of the BLEU scores were at $LR = 1.0$ in all development/test sets. This setting supports no brevity penalty and high n-gram precision. [3]

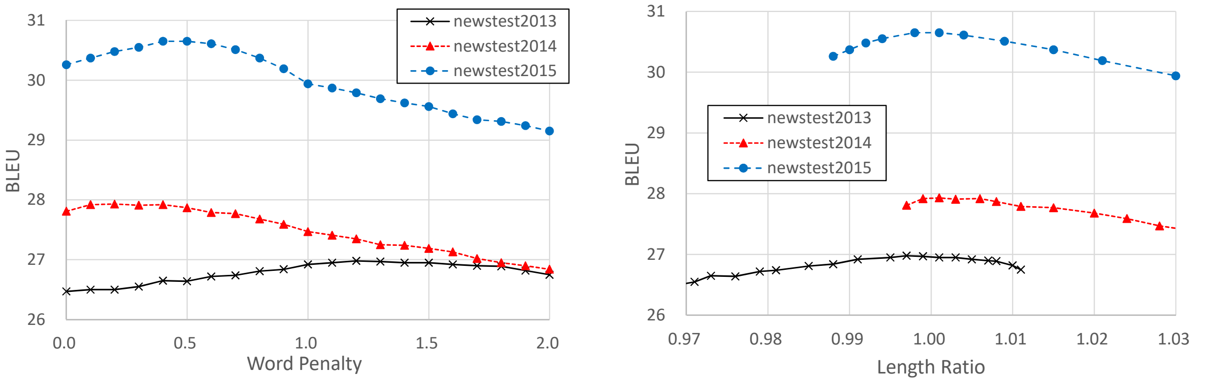
These results reveal that the length ratio should be constant for fair comparison when we compare different systems because they generate translations of different lengths. Therefore, we compare different models and settings by tuning the word penalty to maintain the stable length ratio on the development set (`newstest2013`). In this experiment, we show results of the two length ratios based on the “original parallel corpus only” of the transformer model. Note that the submitted system employs the first setting.

1. $LR \simeq 0.988$, which is the length ratio when $WP = 1.0$.
2. $LR \simeq 0.973$, which is the length ratio when $WP = 0.5$.

4.2 Results

Tables 2 and 3 show the results of Marian NMT (the transformer model) and OpenNMT (the seq-

[3] If we tune the word penalty to make the BLEU score maximum on the `newstest2013`, the length ratios of `newstest2014` and `2015` become greater than 1.0.



(a) Word Penalty vs. BLEU

(b) Length Ratio vs. BLEU

Figure 2: Word Penalty, Length Ratio and BLEU Scores (Marian NMT; $1/\tau = 1.0$)

to-seq model), respectively. These tables consist of three information groups. The first group shows training results; the number of training epochs and perplexity of the development set. The second and third groups show the BLEU scores when the length ratio in the development set become 0.988 and 0.973, respectively. The results of Marian NMT were better than those of OpenNMT in all cases. The following discussion mainly focuses on the results of Marian NMT (Table 2), but there was similarity in Table 3.

First, in comparison with the "original parallel corpus only" and the "one-best without dynamic generation," the perplexity of the one-best case increased from 4.37 to 4.43. Along with increasing the perplexity, the BLEU scores of the test sets (newstest2014 and newstest2015) degraded to 26.19 and 28.49 when $LR \simeq 0.988$. This result indicates that the self-training, which simply uses one-best translation result, is not effective.

On the contrary, using our self-training method, the perplexities were decreased and the BLEU scores improved significantly regardless of the inverse temperature parameters in most cases. [4] For example, when $1/\tau = 1.0$, the perplexity were decreased to 4.20, and the BLEU scores improved to 27.59 and 30.19 on the newstest2014 and 2015, respectively, when $LR \simeq 0.988$. When $LR \simeq 0.973$, the BLEU scores further improved, but the improvements come from the length ratio. The same tendency was observed in OpenNMT. We can conclude that the proposed self-training method is effective for the transformer and seq-to-seq models.

The effectiveness of the inverse temperature parameter is still unclear because the BLEU scores were depend on the parameters.

5 Conclusions

The self-training method in this paper improves the accuracy without changing the model structure. The experimental results show that the proposed method is effective for both the transformer and seq-to-seq models. Although our self-training method increases training time by more than double, we believe that it is effective for the tasks that emphasize on translation speed because it does not change the translation efficiency.

In this paper, only restricted settings were tested. We require further experiments such as another back-translation methodology and settings of the inverse temperature parameters.

Acknowledgments

This work was supported by "Promotion of Global Communications Plan — Research and Development and Social Demonstration of Multilingual Speech Translation Technology," a program of the Ministry of Internal Affairs and Communications, Japan.

References

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

Alexandra Birch, Andrew Finch, Minh-Thang Luong, Graham Neubig, and Yusuke Oda. 2018. Findings

[4] The significance test was performed using the multeval tool (Clark et al., 2011) at a significance level of 5% ($p < 0.05$). https://github.com/jhclark/multeval

		Training		BLEU ↑ (Dev. $LR \simeq 0.988$)			BLEU ↑ (Dev. $LR \simeq 0.973$)		
		#Epoch	PPL ↓	2013	2014	2015	2013	2014	2015
Original Parallel Corpus Only		49	4.37	26.03	26.81	29.35	25.77	26.95	29.67
One-best w/o Dynamic Generation		61	4.43	26.17	26.19 (-)	28.49 (-)	25.94	26.43 (-)	28.91 (-)
Self-Training	$1/\tau = 1.0$	83	4.20	**26.84** (+)	**27.59** (+)	**30.19** (+)	26.65 (+)	27.92 (+)	30.65 (+)
	$1/\tau = 1.2$	112	4.21	27.01 (+)	27.70 (+)	29.80	26.67 (+)	27.92 (+)	30.00
	$1/\tau = 1.5$	98	4.25	26.75 (+)	27.74 (+)	30.17 (+)	26.51 (+)	28.04 (+)	30.27 (+)

Table 2: Results of Marian NMT (Transformer Model)
The bold values denote the results of the submitted system. (+) and (-) symbols denote results that are significantly improved and degraded from the "original parallel corpus only," respectively.

		Training		BLEU ↑ (Dev. $LR \simeq 0.988$)			BLEU ↑ (Dev. $LR \simeq 0.973$)		
		#Epoch	PPL ↓	2013	2014	2015	2013	2014	2015
Original Parallel Corpus Only		20	5.58	23.45	22.69	25.70	23.23	22.94	25.96
One-best w/o Dynamic Generation		20	5.75	22.86 (-)	22.36	24.35 (-)	N/A (No Word Penalty)		
Self-Training	$1/\tau = 1.0$	20	5.34	23.56	23.15 (+)	26.03	23.38	23.49 (+)	26.33
	$1/\tau = 1.2$	20	5.46	23.59	22.95	25.71	23.38	23.15	25.96
	$1/\tau = 1.5$	20	5.41	23.58	23.13 (+)	26.32 (+)	23.55	23.45 (+)	26.50 (+)

Table 3: Results of OpenNMT (Seq-to-Seq Model)
(+) and (-) symbols denote results that are significantly improved and degraded from the "original parallel corpus only," respectively.

of the second workshop on neural machine translation and gen eration. In *Proceedings of the Second Workshop on Neural Machine Translation and Generation (NMT-2018)*, Melbourne, Australia.

Jonathan H. Clark, Chris Dyer, Alon Lavie, and Noah A. Smith. 2011. Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 176–181, Portland, Oregon, USA.

Kenji Imamura, Atsushi Fujita, and Eiichiro Sumita. 2018. Enhancement of encoder and attention using target monolingual corpora in neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation (NMT-2018)*, Melbourne, Australia.

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++. *arXiv preprint arXiv:1804.00344*.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Vancouver, Canada.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL-2018)*, Melbourne, Australia.

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 (WAT2017)*, pages 89–94, Taipei, Taiwan.

Yusuke Oda, Katsuhito Sudoh, Satoshi Nakamura, Masao Utiyama, and Eiichiro Sumita. 2017. A simple and strong baseline: NAIST-NICT neural machine translation system for WAT2017 English-Japanese translation task. In *Proceedings of the 4th Workshop on Asian Translation (WAT2017)*, pages 135–139, Taipei, Taiwan.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL-2016, Volume 1: Long Papers)*, pages 86–96, Berlin, Germany.

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

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of Advances in Neural Information Processing Systems 27 (NIPS 2014)*, 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. *CoRR*, abs/1706.03762.