Tokyo Metropolitan University Neural Machine Translation System for WAT 2017

Yukio Matsumura and **Mamoru Komachi**
Tokyo Metropolitan University
Tokyo, Japan
matsumura-yukio@ed.tmu.ac.jp
komachi@tmu.ac.jp

Abstract

In this paper, we describe our neural machine translation (NMT) system, which is based on the attention-based NMT (Luong et al., 2015) and uses long short-term memories (LSTM) as RNN. We implemented beam search and ensemble decoding in the NMT system. The system was tested on the 4th Workshop on Asian Translation (WAT 2017) (Nakazawa et al., 2017) shared tasks. In our experiments, we participated in the scientific paper subtasks and attempted Japanese-English, English-Japanese, and Japanese-Chinese translation tasks. The experimental results showed that implementation of beam search and ensemble decoding can effectively improve the translation quality.

1 Introduction

Recently, neural machine translation (NMT) has gained popularity in the field of machine translation. The conventional encoder-decoder NMT (Sutskever et al., 2014; Cho et al., 2014) uses two recurrent neural networks (RNN); one is an encoder, which encodes a source sequence into a fixed-length vector; the other is a decoder, which decodes this vector into a target sequence. Attention-based NMT (Bahdanau et al., 2015; Luong et al., 2015) can predict output words by using the weights of each hidden state of the encoder as the context vector, thereby improving the adequacy of the translation.

Despite the success of attention-based models, several open questions remain in NMT. In general, a unique output word is predicted at each time step. Therefore, if a wrong word is predicted, subsequent words will not be correctly output. To enable better predictions, best practices such as beam search and ensemble decoding are recommended to improve the robustness of the predictions. Beam search keeps better hypotheses during decoding, while ensemble decoding reduces the variance of output during decoding.

In this paper, we describe the NMT system that was tested on the shared tasks at 4th Workshop on Asian Translation (WAT 2017) (Nakazawa et al., 2017). We implemented beam search and ensemble decoding in our NMT system. We applied our NMT system to Japanese-English, English-Japanese, and Japanese-Chinese scientific paper translation subtasks. The experimental results show that beam search and ensemble decoding improve the translation accuracy by 3.55 points in Japanese-English translation and 3.28 points in English-Japanese translation in terms of BLEU (Papineni et al., 2002) scores.

2 Neural Machine Translation

Herein, we describe the architecture of our NMT system as shown in Figure 1. The designed system is based on the attention-based NMT (Luong et al., 2015) and uses long short-term memories (LSTM) as RNN. Our NMT system comprises mainly two components:

- Encoder : one-layer bi-directional LSTM
- Decoder : one-layer uni-directional LSTM

2.1 Encoder

The source sentence is converted into a sequence of one-hot word vectors ($\boldsymbol{X} = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_{|\boldsymbol{X}|}]$) where $|\boldsymbol{X}|$ is the length of source sentence.

At each time step i, the source word embedding vector $\boldsymbol{e}_i^s$ is computed by the following equation.

$$\boldsymbol{e}_i^s = \tanh(\boldsymbol{W}_x \boldsymbol{x}_i) \tag{1}$$

Proceedings of the 4th Workshop on Asian Translation, pages 160–166,
Taipei, Taiwan, November 27, 2017. ©2017 AFNLP

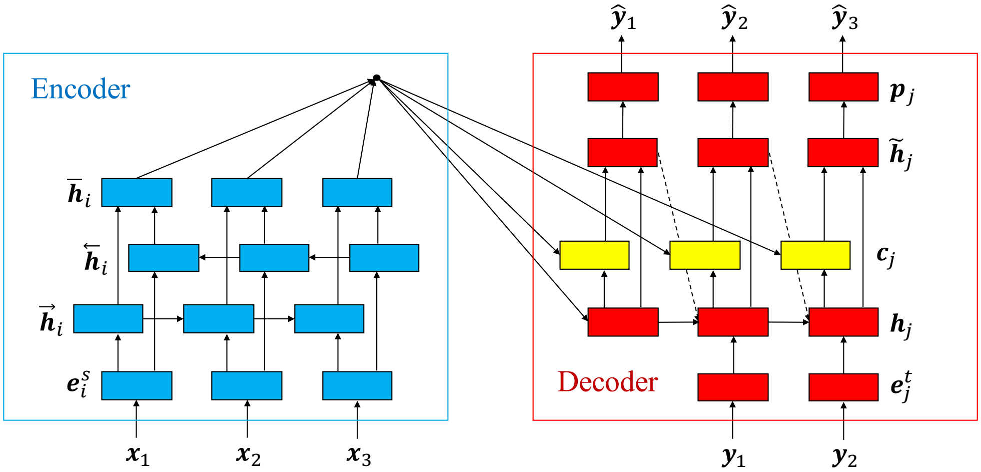

Figure 1: The architecture of our NMT system.

where $\boldsymbol{W}_x \in \mathbb{R}^{q \times v_s}$ is a weight matrix. q is the dimension of the word embeddings and v_s is the size of source vocabulary.

The hidden state $\bar{\boldsymbol{h}}_i$ of the encoder is computed as given by the following equation.

$$\bar{\boldsymbol{h}}_i = \overrightarrow{\boldsymbol{h}_i} + \overleftarrow{\boldsymbol{h}_i}. \tag{2}$$

Here, the forward state $\overrightarrow{\boldsymbol{h}_i}$ and the backward state $\overleftarrow{\boldsymbol{h}_i}$ are computed by

$$\overrightarrow{\boldsymbol{h}_i} = \mathrm{LSTM}(\boldsymbol{e}_i^s, \overrightarrow{\boldsymbol{h}_{i-1}}) \tag{3}$$

and

$$\overleftarrow{\boldsymbol{h}_i} = \mathrm{LSTM}(\boldsymbol{e}_i^s, \overleftarrow{\boldsymbol{h}_{i+1}}). \tag{4}$$

Note that the computation of hidden state $\bar{\boldsymbol{h}}_i$ of the encoder can be regarded as an addition instead of a concatenation.

2.2 Decoder

As with the source sentence, the target sentence is converted into a sequence of one-hot word vectors ($\boldsymbol{Y} = [\boldsymbol{y}_1, \cdots, \boldsymbol{y}_{|\boldsymbol{Y}|}]$) where $|\boldsymbol{Y}|$ is the length of target sentence.

At each time step j, the hidden state $\boldsymbol{h}_j$ of the decoder is represented as

$$\boldsymbol{h}_j = \mathrm{LSTM}([\boldsymbol{e}_{j-1}^t : \tilde{\boldsymbol{h}}_{j-1}], \boldsymbol{h}_{j-1}) \tag{5}$$

where $\boldsymbol{e}_{j-1}^t$ is the target word embedding vector, $\tilde{\boldsymbol{h}}_{j-1}$ is the attentional hidden state, and $\boldsymbol{h}_{j-1}$ is the hidden state at the previous time step.

The target word embedding vector $\boldsymbol{e}_j^t$ is computed by

$$\boldsymbol{e}_j^t = \tanh(\boldsymbol{W}_y \boldsymbol{y}_j) \tag{6}$$

where $\boldsymbol{W}_y \in \mathbb{R}^{q \times v_t}$ is a weight matrix. v_t is the target vocabulary size. The attentional hidden state $\tilde{\boldsymbol{h}}_j$ is represented as

$$\tilde{\boldsymbol{h}}_j = \tanh(\boldsymbol{W}_a[\boldsymbol{h}_j : \boldsymbol{c}_j] + \boldsymbol{b}_a) \tag{7}$$

where $\boldsymbol{W}_a \in \mathbb{R}^{r \times 2r}$ is a weight matrix and $\boldsymbol{b}_a \in \mathbb{R}^r$ is a bias vector. r is the number of hidden units.

The context vector $\boldsymbol{c}_j$ is a weighted sum of each hidden state $\bar{\boldsymbol{h}}_i$ of the encoder. It is represented as

$$\boldsymbol{c}_j = \sum_{i=1}^{|\boldsymbol{X}|} \alpha_{ij} \bar{\boldsymbol{h}}_i. \tag{8}$$

Its weight α_{ij} is a normalized probability distribution, which is computed using a dot product of hidden states, as follows:

$$\alpha_{ij} = \frac{\exp(\bar{\boldsymbol{h}}_i^{\mathrm{T}} \boldsymbol{h}_j)}{\sum_{k=1}^{|\boldsymbol{X}|} \exp(\bar{\boldsymbol{h}}_k^{\mathrm{T}} \boldsymbol{h}_j)}. \tag{9}$$

The conditional probability of the output word $\hat{\boldsymbol{y}}_j$ is computed by

$$\boldsymbol{p}(\hat{\boldsymbol{y}}_j | \boldsymbol{Y}_{<j}, \boldsymbol{X}) = \mathrm{softmax}(\boldsymbol{W}_p \tilde{\boldsymbol{h}}_j + \boldsymbol{b}_p) \tag{10}$$

where $\boldsymbol{W}_p \in \mathbb{R}^{v_t \times r}$ is a weight matrix and $\boldsymbol{b}_p \in \mathbb{R}^{v_t}$ is a bias vector.

Incidentally, the rare words that did not fit in the vocabulary are replaced with unknown tokens "<unk>". When the unknown word is predicted, our NMT system does not process it and outputs this unknown token as it is.

	Japenese-English	Japanese-Chinese
train	1,456,278	672,315
dev	1,790	2,741
test	1,812	2,300

Table 1: Numbers of parallel sentences.

2.3 Training

The objective function is defined by

$$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{D}\sum_{d=1}^{D}\sum_{j=1}^{|\boldsymbol{Y}|}\log \boldsymbol{p}(\boldsymbol{y}_j^{(d)}|\boldsymbol{Y}_{<j}^{(d)}, \boldsymbol{X}^{(d)}, \boldsymbol{\theta}) \tag{11}$$

where D is the number of data and $\boldsymbol{\theta}$ are the model parameters. On training, this objective function is maximized. The model parameters of word embedding are initialized using Word2Vec (Mikolov et al., 2013). The other model parameters are randomly initialized.

2.4 Testing

In general, a unique output word is predicted at each time step. Then the next output word is predicted on the premise that this unique output word is correct. Therefore, if a wrong word is once predicted, then it is difficult to correctly output subsequent words. To make better predictions, we implemented beam search and ensemble decoder.

2.4.1 Beam Search

In general, the word that has the highest probability is output. In beam search, we keep hypotheses of beam size n at each time step. At the subsequent time step, for each hypothesis, we compute n hypotheses; then, we keep n hypotheses in total n^2 hypotheses. Adopting this approach reduces the risk of generating wrong sentences.

2.4.2 Ensemble Decoding

In ensemble decoding, the conditional probability of the output word $\hat{\boldsymbol{y}}_j$ is the average of each model's score. It is computed by

$$\boldsymbol{p}(\hat{\boldsymbol{y}}_j|\boldsymbol{Y}_{<j}, \boldsymbol{X}) = \frac{1}{M}\sum_{m=1}^{M}\boldsymbol{p}^{(m)}(\hat{\boldsymbol{y}}_j|\boldsymbol{Y}_{<j}, \boldsymbol{X}) \tag{12}$$

where M is the number of models. Adopting this approach reduces the risk of predicting a wrong word at each time step.

3 Experiments

We experimented our NMT system on Japanese-English, English-Japanese, and Japanese-Chinese scientific paper translation subtasks.

3.1 Datasets

We used the Japanese-English and Japanese-Chinese parallel corpora in Asian Scientific Paper Excerpt Corpus (ASPEC) (Nakazawa et al., 2014). As regards the Japanese-English parallel corpus, Japanese sentences were segmented by the morphological analyzer MeCab[1] (version 0.996, IPADIC) and English sentences were tokenized by tokenizer.perl of Moses[2]. On the other hand, as regards the Japanese-Chinese parallel corpus, Japanese and Chinese sentences were tokenized by SentencePiece[3]. The vocabulary size of the tokenizer was set to 50,000.

As regards the training data in Japanese-English parallel corpus, we used only the first 1.5 million sentences sorted by sentence-alignment similarity; sentences with more than 60 words were excluded. On the other hand, as regards the training data in Japanese-Chinese parallel corpus, we used all the sentences. Table 1 shows the numbers of the sentences in each parallel corpus.

3.2 Japanese-English and English-Japanese translation tasks

Settings In these tasks, we conducted the experiment using the following configuration:

- Number of hidden units: 1,024
- Word embedding dimensionality: 512
- Source vocabulary size: 100,000
- Target vocabulary size: 30,000
- Minibatch size: 128
- Optimizer: Adagrad
- Initial learning rate: 0.01
- Dropout rate: {0.1, 0.2, 0.3, 0.4, 0.5}
- Beam size: {1, 2, 5, 10, 20}

[1] https://github.com/taku910/mecab
[2] http://www.statmt.org/moses/
[3] https://github.com/google/sentencepiece

Japanese-English				
Model	BLEU	RIBES	AMFM	HUMAN
Previous system (Yamagishi et al., 2016)	18.45	0.711542	0.546880	-
beam 1	21.00	0.725284	0.585710	+56.750
beam 2	22.21	0.733571	0.591740	-
beam 5	22.85	0.737631	0.595180	-
beam 10	22.99	0.739629	0.595030	-
beam 20	23.03	0.741175	0.595260	+61.000
5 ensemble + beam 1	22.78	0.738325	0.587630	-
5 ensemble + beam 2	24.02	0.743581	0.596840	-
5 ensemble + beam 5	24.46	**0.744955**	**0.597760**	-
5 ensemble + beam 10	**24.55**	0.744928	0.596360	-

Table 2: Japanese-English translation results.

English-Japanese				
Model	BLEU	RIBES	AMFM	HUMAN
beam 1	33.72	0.811057	0.740620	+50.750
beam 2	34.54	0.817303	0.744730	-
beam 5	35.10	0.820389	0.744370	-
beam 10	35.30	0.821341	0.744660	-
beam 20	35.32	0.821563	0.744890	+56.500
5 ensemble + beam 1	35.63	0.825683	**0.751660**	-
5 ensemble + beam 2	36.35	0.829732	0.750950	-
5 ensemble + beam 5	36.90	0.831559	0.750360	-
5 ensemble + beam 10	**37.00**	**0.832569**	0.749410	-

Table 3: English-Japanese translation results.

We trained five models with different dropout rates for each task. Then, we selected the best model based on the development set for a single model. The best dropout rate of 0.2 was achieved in a preliminary experiment. We applied various beam sizes during testing. In addition, we ensembled five trained models.

Results Tables 2 and 3 show the translation accuracy in BLEU (Papineni et al., 2002), RIBES (Isozaki et al., 2010), AMFM (Banchs and Li, 2011) and HUMAN evaluation scores. In the "Model" column, "beam n" indicates the model with the beam size of n, "n ensemble" indicates the model ensembled by n trained models on testing. "Previous system" in Table 2 indicates our previous NMT system for WAT 2016 (Yamagishi et al., 2016). This system is based on the attention-based NMT (Bahdanau et al., 2015) and did not implement dropout, beam search, and ensemble decoding.

The results show that beam search and ensemble decoding improve the translation accuracy by 3.55 points in Japanese-English translation and 3.28 points in English-Japanese translation in BLEU scores. As regards Japanese-English translation, our NMT system improved the translation accuracy by 6.10 points compared with our previous NMT system. From a BLEU score standpoint, with increasing beam size, the translation accuracy is enhanced. However, it does not always improve translation accuracy in other metrics.

Table 4 shows examples of outputs of Japanese-English translations. In Example 1, the output is significantly poor when the beam size is 1. However, by increasing the beam size, the output is improves significantly. In Example 2, increasing the beam size does not improve the output; however, by ensemble decoding, the output is improved. The experimental results indicate that beam search and ensemble decoding can effectively improve the translation quality.

Example 1	
Source	単純 桁 橋 より 接合 金具 を 始め 多種 部材 を 組合せる ため , 工法 が 複雑 で ある 。
beam1	since a joint metal
beam20	the method is complicated in order to combine a joint metal metal fitting to a simple girder bridge and a lot of member .
5ensemble + beam10	the method is complicated in order to combine various kinds of members from simple girder bridges to combine various kinds of members .
Reference	the construction was more complicated than simple girder bridge because of combinating various members including connecters .
Example 2	
Source	小型 甲殻 類 で は , アミ 類 の アカイソアミ , ワレカラ 類 の ニッポンワレカラ と ツガルワレカラ は 茨城 県 で 初めて 確認 さ れ た 。
beam1	<unk> , <unk> , <unk> , <unk> , <unk> , <unk> , <unk> , <unk> , <unk> , <unk> , <unk> , <unk> , <unk> , <unk> , <unk> , <unk> , <unk> , <unk> , <unk> , <unk> , <unk> , <unk> , <unk> , <unk> , <unk> , <unk> , <unk> and <unk> , <unk> and <unk> ,
beam20	<unk> , <unk> and <unk> of <unk> , <unk> , <unk> , <unk> , <unk> , <unk> , <unk> , <unk> , <unk> , <unk> , <unk> , <unk> , <unk> , <unk> , <unk> , <unk> , <unk> , <unk> , <unk> , <unk> , <unk> and <unk> , respectively , in Ibaraki Prefecture , for the first time .
5ensemble + beam10	in small crustaceans , <unk> and <unk> of <unk> and <unk> were confirmed for the first time in Ibaraki Prefecture .
Reference	among the small-type Crustacea , Paracanthomysis hispida of Mysidae , and Caprella japonica and C. tsugarensis of Caprellidae were confirmed for the first time in Ibaraki Prefecture .

Table 4: Examples of outputs of Japanese-English translation.

3.3 Japanese-Chinese translation task

Settings In this task, we conducted the experiment using the following configuration:

- Number of hidden units: 1,024
- Word embedding dimensionality: 1,024
- Source vocabulary size: 30,000
- Target vocabulary size: 30,000
- Minibatch size: 64
- Optimizer: Adagrad
- Initial learning rate: 0.01
- Dropout rate: 0.1
- Beam size: 1

Japanese-Chinese			
BLEU	RIBES	AMFM	HUMAN
22.92	0.798681	0.700030	+4.250

Table 5: Japanese-Chinese translation result.

Results Table 5 shows the translation accuracy in terms of BLEU, RIBES, AMFM, and HUMAN evaluation scores. The experimental result indicates that the translation quality is significantly poor compared with the other NMT systems in this task at WAT 2017. As regards this task, because this research is in its infancy, so we could not apply the proper settings. Therefore, we will attempt to pre- or post-process a corpus properly, tune the hyper parameters, and improve the translation quality.

4 Conclusion

In this paper, we described our NMT system, which is based on the attention-based NMT and uses long short-term memories as RNN. We evaluated our NMT system on Japanese-English, English-Japanese, and Japanese-Chinese scientific paper translation subtasks at WAT 2017. The experimental results show that the implementation of beam search and ensemble decoding can effectively improve the translation quality.

In our future work, we will attempt to use the byte pair encoding (BPE) (Sennrich et al., 2016) and compare it with SentencePiece that was explored in this work. In addition, we plan to implement the adversarial NMT (Wu et al., 2017; Yang et al., 2017), which is based on generative adversarial networks (GAN). GAN consist of two networks; one is a discriminator, which distinguishes whether the input data is real or not; the other is a generator, which generates the data that the discriminator cannot distinguish. This approach attempts to generate high quality translations that are comparable to human translations.

Acknowledgement

We express our sincere gratitude to Hayahide Yamagishi, Satoru Katsumata, Michiki Kurosawa, Hiroki Shimanaka, and Longtu Zhang for their support in the tuning of hyper parameters.

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 (ICLR2015)*.

Rafael E Banchs and Haizhou Li. 2011. AM-FM: A Semantic Framework for Translation Quality Assessment. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 153–158, Portland, Oregon, USA. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Hideki Isozaki, Tsutomu Hirao, Kevin Duh, Katsuhito Sudoh, and Hajime Tsukada. 2010. Automatic Evaluation of Translation Quality for Distant Language Pairs. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 944–952, Cambridge, Massachusetts. Association for Computational Linguistics.

Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective Approaches to Attention-based Neural Machine Translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In C J C Burges, L Bottou, M Welling, Z Ghahramani, and K Q Weinberger, editors, *Advances in Neural Information Processing Systems 26 (NIPS2013)*, pages 3111–3119. Curran Associates, Inc.

Toshiaki Nakazawa, Shohei Higashiyama, Chenchen Ding, Hideya Mino, Isao Goto, Graham Neubig, Hideto Kazawa, Yusuke Oda, Jun Harashima, and Sadao Kurohashi. 2017. Overview of the 4th Workshop on Asian Translation. In *Proceedings of the 4th Workshop on Asian Translation (WAT2017)*, Taipei, Taiwan.

Toshiaki Nakazawa, Manabu Yaguchi, Kiyotaka Uchimoto, Masao Utiyama, Eiichiro Sumita, Sadao Kurohashi, and Hitoshi Isahara. 2014. ASPEC : Asian Scientific Paper Excerpt Corpus. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC)*, pages 2204–2208.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

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.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to Sequence Learning with Neural Networks. In Z Ghahramani, M Welling, C Cortes, N D Lawrence, and K Q Weinberger, editors, *Advances in Neural Information Processing Systems 27 (NIPS2014)*, pages 3104–3112. Curran Associates, Inc.

Lijun Wu, Yingce Xia, Li Zhao, Fei Tian, Tao Qin, Jianhuang Lai, and Tie-yan Liu. 2017. Adversarial Neural Machine Translation. *arXiv*, abs/1704.06933.

Hayahide Yamagishi, Shin Kanouchi, Takayuki Sato, and Mamoru Komachi. 2016. Controlling the Voice of a Sentence in Japanese-to-English Neural Machine Translation. In *Proceedings of the 3rd Workshop on Asian Translation (WAT2016)*, pages 203–210, Osaka, Japan. The COLING 2016 Organizing Committee.

Zhen Yang, Wei Chen, Feng Wang, and Bo Xu. 2017. Improving Neural Machine Translation with Conditional Sequence Generative Adversarial Nets. *arXiv*, abs/1703.0.