XMU Neural Machine Translation Systems for WAT 2017

Boli Wang, Zhixing Tan, Jinming Hu, Yidong Chen and **Xiaodong Shi***
School of Information Science and Engineering, Xiamen University, Fujian, China
{boliwang, playinf, todtom}@stu.xmu.edu.cn
{ydchen, mandel}@xmu.edu.cn

Abstract

This paper describes the Neural Machine Translation systems of Xiamen University for the shared translation tasks of WAT 2017. Our systems are based on the Encoder-Decoder framework with attention. We participated in three subtasks. We experimented subword segmentation, synthetic training data and model ensembling. Experiments show that all these methods can give substantial improvements.

1 Introduction

Neural Machine Translation (NMT) (Bahdanau et al., 2015; Cho et al., 2014; Sutskever et al., 2014) has achieved great success in recent years and outperforms traditional statistical machine translation (SMT) on various language pairs (Sennrich et al., 2016a; Wu et al., 2016; Zhou et al., 2016). This paper describes the NMT systems of Xiamen University (XMU) for the WAT 2017 evaluation (Nakazawa et al., 2017). We participated in three translation subtasks: JIJI Japanese↔English newswire subtask, IITB Hindi↔English mixed domain subtasks, and Cookpad Japanese↔English recipe subtask.

In all three subtasks, we use our reimplementation of dl4mt-tutorial[1] with minor changes. We use both Byte Pair Encoding (BPE) (Sennrich et al., 2016c) and mixed word/character segmentation (Wu et al., 2016) to achieve open-vocabulary translation. We apply back-translation method (Sennrich et al., 2016b) to make use of monolingual data. We use ensemble (Sutskever et al., 2014) of multiple models to further improve the translation quality.

The remainder of this paper is organized as follows: Section 2 describes our NMT system, including the training details. Section 3 describes the processing of the data. Section 4 describes all experimental features. Section 5 shows the results of our experiments. Finally, we conclude in section 6.

2 Baseline System

Our NMT system is a reimplementation of dl4mt-tutorial model. We import some minor changes and new features such as dropout (Srivastava et al., 2014).

For all three subtasks, we train our models with almost the same settings of hyper-parameters. We use word embeddings of size 620 and hidden layers of size 1000. We use mini-batches of size 128 and adopt Adam (Kingma and Ba, 2015) ($\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 1 \times 10^{-8}$) as the optimizer. The initial learning rate is set to 5×10^{-4}. We gradually halve the learning rate during the training process. As a common way to train RNN models, we clip the norm of gradients to a predefined value 1.0 (Pascanu et al., 2013). We use dropout to avoid over-fitting with a keep probability of 0.8. For ensembling, we train multiple models with different random initialization of parameters and different data shuffling.

In Decoding, we employ beam search strategy with a beam size of 10. We use a modified version of AmuNMT C++ decoder[2] for parallel decoding. We use the same ensembling method as (Sutskever et al., 2014) with uniform weights for different models.

*Corresponding author.

[1] https://github.com/nyu-dl/dl4mt-tutorial

[2] https://github.com/emjotde/amunmt

Proceedings of the 4th Workshop on Asian Translation, pages 95–98,
Taipei, Taiwan, November 27, 2017.

3 Data Processing

We use all training data provided by JIJI, IITB, and Cookpad corpora[3]. For JIJI and Cookpad corpora, Moses[4] tokenizer and truecaser are applied on the English side. On the Japanese side, the full-width ASCII variants are first converted into their half-width form and the `mecab`[5] segmenter is used to segment the sentences. For IITB corpus, we directly use the tokenized data and truecase the English sentences with Moses truecaser.

For all three corpora, we remove duplications and filter out bad sentence pairs according to the word alignment scores obtained by `fast-align` toolkit[6]. For IITB corpus, we also filter out sentence pairs which are not in English-Hindi according to the range of Devanagari characters' Unicode, as well as a language identification toolkit `langid`[7].

4 Experimental Features

4.1 Subword Segmentation

To enable open-vocabulary, we apply subword-based translation approaches. In our preliminary experiments, we found that BPE and mixed word/character segmentation works better than UNK replacement techniques.

In JIJI and IITB tasks, we apply BPE[8] with 20K operations to English sentences and Hindi sentences separately. We use mixed word/character model in the Japanese sides of JIJI task. We keep 20K most frequent Japanese words and split other words into characters. Unlike (Wu et al., 2016), we do not add any extra prefixes or suffixes to the segmented Japanese characters. In the post-processing step, we simply remove all spaces in Japanese sentences.

Similarly, in Cookpad task, we also use BPE segmentation in English side, but with 10K operations, since the vocabulary size is much smaller. Correspondingly, mixed word/character model with a shortlist of 10K words is applied to the Japanese sentences.

[3]For Cookpad corpus, we extract parallel pairs from six fields: *step*, *history*, *ingredient*, *title*, *advice*, and *description*.

[4]http://statmt.org/moses/

[5]https://taku910.github.io/mecab/

[6]https://github.com/clab/fast_align

[7]https://pypi.python.org/pypi/langid

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

4.2 Synthetic Training Data

To utilize the monolingual data in IITB corpus, we employ the back-translation method. We use `srilm`[9] to train a 5-gram KN language model on the monolingual data and select monolingual sentences according to their perplexity. By this way, 2.5M English sentences are selected from IITB's monolingual data. We use one single EN-HI NMT baseline model to translate the selected English monolingual sentences back to Hindi. The synthetic sentence pairs are used to train HI-EN NMT models.

Similarly, we also select 2.5M Hindi monolingual sentences and use one single HI-EN NMT baseline model to translate them back to English. The synthetic sentence pairs are used to train EN-HI NMT models.

In preliminary experiments, we found that training or tuning on the synthetic data alone could not significantly improve the performance of NMT models. Therefore, we mix up the synthetic data with a comparable amount of bilingual pairs over sampled from IITB's parallel data and train NMT models on the mixture data. A similar method is also used in (Sennrich et al., 2017).

5 Results

In this section, we report the automatic evaluation results (word-level BLEU score[10]) and human evaluation results on test sets. We compare our NMT systems with the best SMT systems provided by the organizer.

5.1 Results on JIJI Subtask

System	EN-JA		JA-EN	
	BLEU	Human	BLEU	Human
HPBMT	16.22	10.25	15.67	10.25
Baseline	17.92	– –	15.77	– –
+Ensemble	**20.14**	**11.75**	**17.95**	**20.75**

Table 1: Automatic evaluation and human evaluation results on JIJI subtask.

Table 1 shows the results of JIJI subtask. We apply subword segmentation on the parallel data and train 4 English-Japanese NMT models and 4

[9]http://www.speech.sri.com/projects/srilm/

[10]The references and translations are tokenized by Moses English tokenizer, Mecab Japanese word segmenter and Indic Hindi tokenizer respectively.

Japanese-English models. We found that both in EN-JP and JP-EN, one single NMT model can outperform the traditional SMT systems, such as a hierarchical phrase-based model. Ensembles of 4 NMT models can further improve the results by more than +2.0 BLEU scores.

5.2 Results on IITB Subtask

System	EN-HI BLEU	EN-HI Human	HI-EN BLEU	HI-EN Human
PBMT	10.79	– –	10.32	– –
Baseline	13.69	– –	13.30	– –
+Synthetic	19.79	– –	20.61	– –
+Ensemble	**21.39**	**64.50**	**22.44**	**68.25**

Table 2: Automatic evaluation and human evaluation results on IITB subtask.

In IITB subtask, we first train an English-Hindi and a Hindi-English baseline NMT models on the parallel data with subword segmentation. Then we select monolingual sentences and synthesize larger training data using the backward baseline NMT models. As shown in Table 2, both in EN-HI and HI-EN, training on synthetic data is effective to improve the BLEU score (more than +6.0). When ensembling 4 models, we further gain more than +1.6 BLEU scores.

5.3 Results on Cookpad Subtask

In Cookpad subtask, we hope one single NMT model has the robustness to translate different types of text. So we directly train NMT models on all training data without any extra data separation or labelling. And we use the same models for four test sets. The results are shown in Table 3. Our single NMT baselines beat phrase-based SMTs in almost all test sets, except for JA-EN *ingredient*. When ensembling 4 models, we further gain +1.3 to +3.1 BLEU scores in all test sets and outperform SMTs by +2.2 to +5.8 BLEU scores. For human evaluation results, we found that NMT models achieve good results in *title* and *step* sets, but not in *ingredient* sets. It's reasonable because NMT models are good at fluency, instead of adequacy. And for *title* and *step*, human readers usually focus on fluency. But for *ingredient*, human readers care more about adequacy.

System	EN-JA BLEU	EN-JA Human	JA-EN BLEU	JA-EN Human
all				
PBMT	19.10	– –	23.87	– –
Baseline	22.47	– –	27.04	– –
+Ensemble	**24.44**	– –	**28.83**	– –
title				
PBMT	16.57	– –	9.72	– –
Baseline	16.90	– –	14.25	– –
+Ensemble	**18.78**	**23.75**	**15.57**	**10.25**
step				
PBMT	18.53	– –	22.84	– –
Baseline	22.01	– –	26.31	– –
+Ensemble	**24.00**	**45.50**	**28.03**	**40.50**
ingredient				
PBMT	29.60	– –	44.42	– –
Baseline	30.90	– –	43.89	– –
+Ensemble	**33.19**	**-3.75**	**46.98**	**3.50**

Table 3: Automatic evaluation and human evaluation results on Cookpad subtask.

6 Conclusion

We describe XMU's neural machine translation systems for the WAT 2017 shared translation tasks. Our models perform quite well and proved to be effective enough to outperform traditional SMT systems in all tasks, even with limited training data. Experiments also show the effectiveness of all features we used, including subword segmentation, synthetic training data, and multi-model ensemble.

Acknowledgments

This work was supported by the Natural Science Foundation of China (Grant No. 61573294), the Ph.D. Programs Foundation of Ministry of Education of China (Grant No. 20130121110040), the Foundation of the State Language Commission of China (Grant No. WT135-10) and the National High-Tech R&D Program of China (Grant No. 2012BAH14F03).

References

Dzmitry Bahdanau, KyungHyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of ICLR*.

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 EMNLP*, pages 1724–1734.

Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of ICLR*.

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.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. In *Proceedings of ICML*, pages 1310–1318.

Rico Sennrich, Alexandra Birch, Anna Currey, Ulrich Germann, Barry Haddow, Kenneth Heafield, Antonio Valerio Miceli Barone, and Philip Williams. 2017. The University of Edinburgh's Neural MT Systems for WMT17. *arXiv preprint arXiv:1708.00726*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Edinburgh neural machine translation systems for wmt 16. *arXiv preprint arXiv:1606.02891*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Improving neural machine translation models with monolingual data. In *Proceddings of ACL*, pages 86–96.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016c. Neural machine translation of rare words with subword units. In *Proceedings of 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. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Jie Zhou, Ying Cao, Xuguang Wang, Peng Li, and Wei Xu. 2016. Deep recurrent models with fast-forward connections for neural machine translation. *Transactions of the Association for Computational Linguistics*, 4:371–383.