XMU Neural Machine Translation Systems for WMT 17

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

Abstract

This paper describes the Neural Machine Translation systems of Xiamen University for the translation tasks of WMT 17. Our systems are based on the Encoder-Decoder framework with attention. We participated in three directions of shared news translation tasks: English→German and Chinese↔English. We experimented with deep architectures, different segmentation models, synthetic training data and target-bidirectional translation models. Experiments show that all methods can give substantial improvements.

1 Introduction

Neural Machine Translation (NMT) (Cho et al., 2014; Sutskever et al., 2014; Bahdanau et al., 2015) has achieved great success in recent years and obtained state-of-the-art results on various language pairs (Zhou et al., 2016; Sennrich et al., 2016a; Wu et al., 2016). This paper describes the NMT systems of Xiamen University (XMU) for the WMT 17. We participated in three directions of shared news translation tasks: English→German and Chinese↔English. We use two different NMTs for shared news translation tasks:

- MININMT: A deep NMT system (Zhou et al., 2016; Wu et al., 2016; Wang et al., 2017) with a simple architecture. The decoder is a stacked Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) with 8 layers. The encoder has two variants. For English-German translation, we use an interleaved bidirectional encoder with 2 columns. Each column consists of 4 LSTMs. For Chinese-English translation, we use a stacked bidirectional encoder with 8 layers.

- DL4MT: Our reimplementation of dl4mt-tutorial[1] with minor changes. We also use a modified version of AmuNMT C++ decoder[2] for decoding. This system is used in the English-Chinese translation task.

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. Back-translation method (Sennrich et al., 2016b) is applied to make use of monolingual data. We also use target-bidiretional translation models to alleviate the label bias problem (Lafferty et al., 2001).

The remainder of this paper is organized as follows: Section 2 describes the architecture of MININMT. Section 3 describes all experimental features used in WMT 17 shared translation tasks. Section 4 shows the results of our experiments. Section 5 shows the results of shared translation task. Finally, we conclude in section 6.

2 Model Description

Deep architectures have recently shown promising results on various language pairs (Zhou et al., 2016; Wu et al., 2016; Wang et al., 2017). We also experimented with a deep architecture as depicted in Figure 1. We use LSTM as the main recurrent unit and residual connections (He et al., 2016) to help training.

Given a source sentence $\boldsymbol{x} = \{x_1, \ldots, x_S\}$ and a target sentence $\boldsymbol{y} = \{y_1, \ldots, y_T\}$, the encoder maps the source sentence $\boldsymbol{x}$ into a sequence of annotation vectors $\{\mathbf{x}_i\}$. The decoder produces

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

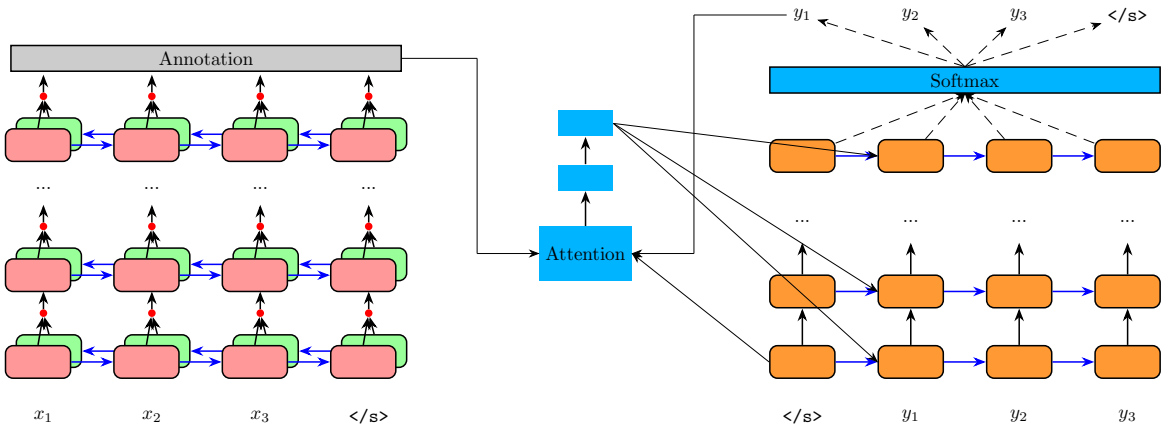

Figure 1: The architecture of our deep NMT system, which is inspired by Deep-Att (Zhou et al., 2016) and GNMT (Wu et al., 2016). Both the encoder and decoder adopt LSTM as its main recurrent unit. We also use residual connections (He et al., 2016) to help training, but here we omit it for clarity. We use black lines to denote input connections while use blue lines to denote recurrent connections.

translation y_t given the source annotation vectors $\{\mathbf{x}_i\}$ and target history $\boldsymbol{y}_{<t}$.

2.1 Encoder

2.1.1 Interleaved Bidirectional Encoder

The interleaved bidirectional encoder was introduced by (Zhou et al., 2016), which is also used in (Wang et al., 2017). Like (Zhou et al., 2016), our interleaved bidirectional encoder consists of two columns. In interleaved bidirectional encoder, the LSTMs in adjacent layers run in opposite directions:

$$\overrightarrow{\mathbf{x}}_t^i = \text{LSTM}_i^f(\overrightarrow{\mathbf{x}}_t^{i-1}, \overrightarrow{\mathbf{s}}_{t+(-1)^i}^i) \quad (1)$$

$$\overleftarrow{\mathbf{x}}_t^i = \text{LSTM}_i^b(\overleftarrow{\mathbf{x}}_t^{i-1}, \overleftarrow{\mathbf{s}}_{t+(-1)^{i+1}}^i) \quad (2)$$

Here $\mathbf{x}_t^0 \in \mathbb{R}^e$ is the word embedding of word x_t, $\mathbf{x}_t^i \in \mathbb{R}^h$ is the output of LSTM unit and $\mathbf{s}_t^i = (\mathbf{c}_t^i, \mathbf{m}_t^i)$ denotes the memory and hidden state of LSTM. We set both e and h to 512 in all our experiments. The annotation vectors $\mathbf{x}_i \in \mathbb{R}^{2h}$ are obtained by concatenating the final output $\overrightarrow{\mathbf{x}}^{L_{\text{enc}}}$ and $\overleftarrow{\mathbf{x}}^{L_{\text{enc}}}$ of two encoder columns. In our experiments, we set $L_{\text{enc}} = 4$.

2.1.2 Stacked Bidirectional Encoder

To better exploit source representation, we adopt a stacked bidirectional encoder. As shown in Figure 1, all layers in the encoder are bidirectional. The calculation is described as follows:

$$\overrightarrow{\mathbf{x}}^i = \text{LSTM}_i^f(\mathbf{x}_t^{i-1}, \overrightarrow{\mathbf{s}}_{t-1}^i) \quad (3)$$

$$\overleftarrow{\mathbf{x}}^i = \text{LSTM}_i^b(\mathbf{x}_t^{i-1}, \overleftarrow{\mathbf{s}}_{t+1}^i) \quad (4)$$

$$\mathbf{x}^i = [\overrightarrow{\mathbf{x}}^{iT}; \overleftarrow{\mathbf{x}}^{iT}]^T \quad (5)$$

To reduce parameters, we reduce the dimension of hidden units from h to $h/2$ so that $\mathbf{x}^i \in \mathbb{R}^h$. The annotation vectors are taken from the output $\mathbf{x}^{L_{\text{enc}}}$ of top LSTM layer. In our experiments, L_{enc} is set to 8.

2.2 Decoder

The decoder network is similar to GNMT (Wu et al., 2016). At each time-step t, let $\mathbf{y}_{t-1}^0 \in \mathbb{R}^e$ denotes the word embedding of y_{t-1} and $\mathbf{y}_{t-1}^1 \in \mathbb{R}^h$ denotes the output of bottom LSTM from previous time-step. The attention network calculates the context vector $\mathbf{a}_t$ as the weighted sum of source annotation vectors:

$$\mathbf{a}_t = \sum_{i=1}^{S} \alpha_{t,i} \cdot \mathbf{x}_i \quad (6)$$

Different from GNMT (Wu et al., 2016), we use the concatenation of $\mathbf{y}_{t-1}^0$ and $\mathbf{y}_{t-1}^1$ as the query vector for attention network, as described follows:

$$\mathbf{h}_t = [\mathbf{y}_{t-1}^{0\,T}; \mathbf{y}_{t-1}^{1\,T}]^T \quad (7)$$

$$e_{t,i} = \mathbf{v}_a^T \tanh(\mathbf{W}_a \mathbf{h}_t + \mathbf{U}_a \mathbf{x}_i) \quad (8)$$

$$\alpha_{t,i} = \frac{\exp(e_{t,i})}{\sum_{j=1}^{S} \exp(e_{t,j})} \quad (9)$$

This approach is also used in (Wang et al., 2017). The context vector $\mathbf{a}_t$ is then fed to all decoder LSTMs.

The probability of the next word y_t is simply modeled using a softmax layer on the output of top LSTM:

$$p(y_t|\boldsymbol{x}, \boldsymbol{y}_{<t}) = \text{softmax}(y_t, \mathbf{y}_t^{L_{\text{dec}}}) \quad (10)$$

We set L_{dec} to 8 in all our experiments.

3 Experimental Features

3.1 Segmentation Approaches

To enable open-vocabulary, we use two approaches: BPE and mixed word/character segmentation.

In most of our experiments, we use BPE[3] (Sennrich et al., 2016c) with 50K operations. In our preliminary experiments, we found that BPE works better than UNK replacement techniques.

For English-Chinese translation task, we apply mixed word/character model (Wu et al., 2016) to Chinese sentences. We keep the most frequent 50K words and split other words into characters. Unlike (Wu et al., 2016), we do not add any prefixes or suffixes to the segmented Chinese characters. In post-processing step, we simply remove all the spaces.

3.2 Synthetic Training Data

We apply back-translation (Sennrich et al., 2016b) method to use monolingual data. For English-German and Chinese-English translation, we sample monolingual data from the NewsCrawl2016 corpora. For English-Chinese translation, we sample monolingual data from the XinhuaNet2011 corpus.

3.3 Target-bidirectional Translation

For Chinese-English translation, we also use a target-bidirectional model (Liu et al., 2016; Sennrich et al., 2016a) to rescore the hypotheses.

To train a target-bidirectional model, we reverse the target side of bilingual pairs from left-to-right (L2R) to right-to-left (R2L). We first output 50 candidates from the ensemble of 4 L2R models. Then we rescore candidates by interpolating L2R score and R2L score with uniform weights.

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

3.4 Training

For all our models, we adopt Adam (Kingma and Ba, 2015) ($\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 1 \times 10^{-8}$) as the optimizer. The learning rate is set to 5×10^{-4}. We gradually halve the learning rate during the training process. As a common way to train RNNs, we clip the norm of gradient to a predefined value 5.0. The batch size is 128. We use dropout (Srivastava et al., 2014) to avoid overfitting with a keep probability of 0.8.

4 Results

4.1 Results on English-German Translation

System	Test (BLEU)
Baseline	25.7
+Synthetic	26.1
+Ensemble	**26.7**

Table 1: English-German translation results on newstest2017.

Table 1 show the results of English-German Translation. The baseline system is trained on preprocessed parallel data[4]. For synthetic data, we randomly sample 10M German sentences from NewsCrawl2016 and translate them back to English using an German-English model. However, we found random sampling do not work well. As a result, for Chinese-English translation, we select monolingual data according to development set. We first train one baseline model and continue to train 4 models on synthetic data with different shuffles. Next we ensemble 4 models and get the final results. We found this approach do not lead to substantial improvements.

4.2 Results on Chinese-English Translation

System	Test (BLEU)
Baseline	23.1
+Synthetic	23.7
+Ensemble	25.3
+R2L reranking	**26.0**

Table 2: Chinese-English translation results on newstest2017.

We use all training data (CWMT Corpus, UN Parallel Corpus and News Commentary) to train a

[4] http://data.statmt.org/wmt17/translation-task/preprocessed/de-en/

baseline system. The Chinese sentences are segmented using Stanford Segmenter[5]. For English sentences, we use the moses tokenizer[6]. We filter bad sentences according to the alignment score obtained by `fast-align` toolkit[7] and remove duplications in the training data. The preprocessed training data consists of 19M bilingual pairs. As noted earlier, the monolingual data is selected using newsdev2017. We first train 4 L2R models and one R2L model on training data, then we finetune our model on a mixture of 2.5M synthetic bilingual pairs and 2.5M bilingual pairs sampled from CWMT corpus. As shown in Table 2, we obtained +1.6 BLEU score when ensembling 4 models. When rescoring with one R2L model, we further gain +0.7 BLEU score.

4.3 Results on English-Chinese Translation

System	Test (BLEU)
Baseline	30.4
+Synthetic	34.3
+Ensemble	**35.8**

Table 3: English-Chinese translation results on newstest2017.

Table 3 show the results of English-Chinese Translation. We use our reimplementation of DL4MT to train English-Chinese models on CWMT and UN parallel corpus. The preprocessing steps, including word segmentation, tokenization, and sentence filtering, are almost the same as Section 4.2, except that we limited the vocabulary size to 50K and split all target side OOVs into characters. For synthetic parallel data, we use SRILM[8] to train a 5-gram KN language model on XinhuaNet2011 and select 2.5M sentences from XinhuaNet2011 according to their perplexities. We obtained +3.9 BLEU score when tuning the single best model on a mixture of 2.5M synthetic bilingual pairs and 2.5M bilingual pairs selected from CWMT parallel data randomly. We further gain +1.5 BLEU score when ensembling 4 models.

[5] https://nlp.stanford.edu/software/segmenter.shtml

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

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

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

5 Shared Task Results

Table 4 shows the ranking of our submitted systems at the WMT17 shared news translation task. Our submissions are ranked (tied) first for 2 out of 3 translation directions in which we participated: EN↔ZH.

Direction	BLEU Rank	Human Rank
EN→DE	4	2-9 of 16
ZH→EN	2	1-3 of 16
EN→ZH	2	1-3 of 11

Table 4: Automatic (BLEU) and human ranking of our submitted systems at WMT17 shared news translation task.

6 Conclusion

We describe XMU's neural machine translation systems for the WMT 17 shared news translation tasks. All our models perform quite well on all tasks we participated. Experiments also show the effectiveness of all features we used.

Acknowledgments

This work was supported by the Natural Science Foundation of China (Grant No. 61573294, 61303082, 61672440), 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 Natural Science Foundation of Fujian Province (Grant No. 2016J05161).

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 the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778.

Sepp Hochreiter and Jurgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, pages 1735–1780.

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

John Lafferty, Andrew McCallum, Fernando Pereira, et al. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the eighteenth international conference on machine learning, ICML*, volume 1, pages 282–289.

Lemao Liu, Masao Utiyama, Andrew Finch, and Eiichiro Sumita. 2016. Agreement on target-bidirectional neural machine translation. In *Proceedings of NAACL-HLT*, pages 411–416.

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*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016c. Neural machine translation of rare words with subword units. In *Proceedings of ACL*.

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.

Mingxuan Wang, Zhengdong Lu, Jie Zhou, and Qun Liu. 2017. Deep Neural Machine Translation with Linear Associative Unit. *arXiv preprint arXiv:1705.00861*.

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.