Tsinghua University Neural Machine Translation Systems for CCMT 2020

Gang Chen[1] ⋆, Shuo Wang[1] ⋆, Xuancheng Huang[1] ⋆, Zhixing Tan[1], Maosong Sun[1,2], and Yang Liu[1,2]

[1] Institute for Artificial Intelligence
Department of Computer Science and Technology, Tsinghua University
Beijing National Research Center for Information Science and Technology
[2] Beijing Academy of Artificial Intelligence

Abstract. This paper describes the neural machine translation system of Tsinghua University for the bilingual translation task of CCMT 2020. We participated in the Chinese↔English translation tasks. Our systems are based on Transformer architectures and we verified that deepening the encoder can achieve better results. All models are trained in a distributed way. We employed several data augmentation methods, including knowledge distillation, back-translation, and domain adaptation, which are all shown to be effective to improve translation quality. Distinguishing original text from translationese can lead to better results when performing domain adaptation. We found model ensemble and transductive ensemble learning can further improve the translation performance over the individual model. In both Chinese→English and English→Chinese translation tasks, our systems achieved the highest case-sensitive BLEU score among all submissions.

1 Introduction

This paper describes the neural machine translation (NMT) systems of Tsinghua University for the CCMT 2020 translation task. We participated in two directions of bilingual translation tasks: Chinese→English and English→Chinese. We exploited the following techniques to build our systems:

- Deep Transformers: We train deep transformer models with mixed-precision and distributed training.
- Data augmentation: We explored various data augmentation methods such as back-translation and knowledge distillation.
- Finetuning and model ensemble: We use finetuning and model ensemble to further improve the performance of our systems.

The overview of our methods is shown in Figure 1. The remainder of this paper is structured as follows: Section 2 describes the methods used in our CCMT 2020 submissions. Section 3 shows the settings and results of our experiments. Finally, we conclude in Section 4.

⋆ Equal contributions. Listing order is random.

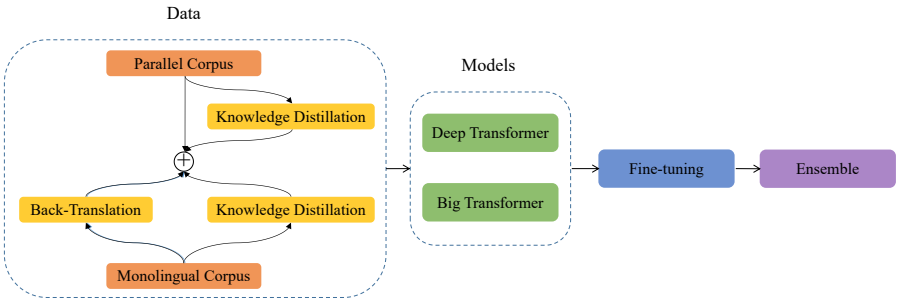

Fig. 1. Overview of Tsinghua NMT systems.

2 Methods

2.1 Data

We use all available data provided by CCMT and WMT, which contains a total of 26.7M bilingual sentence pairs. We apply the following procedures to preprocess the data:

- We remove illegal UTF-8 characters and replace all control characters with a space character.
- All Traditional Chinese sentences are converted into Simplified Chinese ones.
- We apply Unicode NFKC normalization to normalize texts.
- We further restore HTML/XML escape and normalize punctuation in the texts.

The resulting parallel corpus still contains many noise sentence pairs. Therefore, we further filter the data using the following rules:

- We remove all duplicate sentence pairs in the data.
- All sentences that contain illegal characters (e.g., Chinese characters in English sentences) are discarded.
- We translate both the Chinese and English sides of the bilingual data with baseline NMT systems and compute the sentence-level BLEU scores between the translated and original sentences. Then we discard all sentence pairs with BLEU scores below 5.

After filtering, the final data used in our experiments contains about 21M sentence pairs. To tokenize the texts, we use Jieba segmenter [3] for Chinese and Moses toolkit [4] for English.

2.2 Models

Deep Transformer According to the previous work [6], the performance of Transformer models [7] can be improved by increasing the number of layers in the encoder.

[3] https://github.com/fxsjy/jieba
[4] https://github.com/moses-smt/mosesdecoder

We follow [6] to use deep Transformer in our experiments. To address the vanishing-gradient problem in deep Transformer, we adopt the pre-layer normalization [8] instead of the post-layer normalization [7].

In our experiments, both the big Transformer with 15 encoder layers and the base Transformer with 50 encoder layers obtain significant improvements compared with the vanilla big Transformer on Chinese→English translation task.

2.3 Data Augmentation

Back Translation We augment the training data by exploring the monolingual corpus using back translation [4,1]. Specifically, we select a portion of sentences in the target monolingual corpus and then translate them back into the source language using target-to-source (T2S) models. We merge the synthetic data with the bilingual data to train our models. Following Edunov et al. [1], we also add noise to the translated sentences to further improve the performance.

Knowledge Distillation We further augment the data by applying sequence-level knowledge distillation (KD) [2]. We explore the following types of KD in our experiments:

- R2L KD: We replace the target-side sentences of the parallel corpus with sentences translated by Right-to-Left (R2L) models.
- Ensemble KD: We ensemble multiple models to translate the source-side sentences in the parallel corpus.
- Monolingual KD: We exploit additional source-side monolingual data by translating them using existing NMT models.

2.4 Finetuning

Previous work [6] found that finetuning with in-domain data can bring huge improvements. We also use development sets as the in-domain datasets. As mentioned in Sun et al. [6], the source side of `newsdev2017`, `newstest2017` and `newstest2018` are composed of two parts: documents created originally in Chinese and documents created originally in English. We split these datasets into original Chinese part and original English part according to tag attributes of SGM files. For Chinese-English translation, we use CWMT2008, CWMT2009 and original Chinese part of `newsdev2017`, `newstest2017` and `newstest2018` as the in-domain dataset. For English-Chinese translation, we use original English part of `newsdev2017`, `newstest2017` and `newstest2018` as the in-domain dataset.

During finetuning, we use a larger dropout rate and a smaller constant learning rate than those in the training process. The model parameters are updated after each epoch, which is enabled by gradient accumulation. We finetune all models with 18 epochs.

2.5 Ensemble

Model ensemble is a well-known technique to combine different models for stronger performance. We utilize the frequently used method for ensemble, which calculates

Settings	Transformer Big	Deep Transformer Base	Deep Transformer Big
Baseline	27.94	-	-
+Data Augmentation	28.59	29.74	29.85
+Finetuning	35.97	37.48	37.74
Ensemble		**38.95**	

Table 1. BLEU evaluation results on the `newstest2019` Chinese-English test set.

the word-level averaged log-probability among different models during decoding. On account of that the model diversity among single models has a strong impact on the performance of ensemble model, we combine single models that have different model architectures (e.g., different number of encoder layers or different widths of the feed-forward layer) and been trained on different data (e.g., generated by different data augmentation method).

We also try to use Transductive Ensemble Learning (TEL) [9] instead of standard ensemble. TEL is a technique utilizing the synthetic test data (consists of original source sentences and translations of target-side sentences) of different models to finetune a single model. In our experiments, we find that once several single models have been applied TEL, their ensemble model could not outperform single models. We employ 5 left-to-right models and 2 right-to-left models to generate synthetic test data and finetune our best single model. Finally, we get a single model which even outperforms the ensemble of several single models.

3 Experiments

3.1 Settings

We use the PyTorch implementation of open-source toolkit THUMT [5] to carry out all experiments. To enable open vocabulary, we learn 32K BPE [5] operations separately on Chinese and English texts using `subword-nmt` [6] toolkit. All models are trained on 2 machines with 10 RTX 2080Ti GPUs on each machine.

For all our models, we adopt Adam [3] ($\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 1\times 10^{-8}$) as the optimizer. We use the default hyperparameters provided by THUMT to train Transformer models. In addition, we use distributed training and mixed-precision training to reduce the training time. Specifically, we set the batch size to 2048 source and target tokens on each GPU per step, and accumulate the gradients for 10 steps to update the model parameters. We train each model for only 50k steps. It takes about 2 days to train a deep Transformer model. After training, we average 5 top checkpoints validated on the validation set [10] and then perform finetuning on top of this new checkpoint.

3.2 Results on Chinese-English Translation

Table 1 shows the results of Chinese-English Translation on `newstest2019` dataset. All methods we used can bring substantial improvements over the baseline system.

[5] `https://github.com/THUNLP-MT/THUMT`
[6] `https://github.com/rsennrich/subword-nmt`

Dataset	Baseline	+BT	+Noise BT	+BT&EKD	+BT&R2LKD
newstest2019	34.85	34.53	35.05	35.65	35.02

Table 2. Detailed experiments on different data augmentation methods. All BLEU scores are reported after finetuning on development sets.

Dataset	Data Augmentation		Deep model	Finetuning	Ensemble
	BT	Noise BT			
newstest2019	36.94	37.02	36.91	38.33	39.56

Table 3. BLEU evaluation results on the newstest2019 English-Chinese test set.

Applying data augmentation methods improves the baseline system by 0.65 BLEU score. The deep models can further bring 1.26 BLEU improvements. In our experiments, finetuning with in-domain data is the most effective approach, which gains about 7$\sim$8 BLEU improvements. Furthermore, the gap between Transformer Big and Deep Transformer Big model enlarges after applying the finetuning step.

Table 2 shows the detailed results of different data augmentation methods. The results are reported after applying the finetuning step to see whether the methods can bring further improvements. We have the following findings:

- Back translation does not work well on Chinese-English translation. Considering finetuning on texts translated from Chinese is very effective, we conjecture that the result is caused by the mismatch of style between texts originally from English and texts translated from Chinese to English.
- Adding noise to pseudo-source sentences is helpful to improve translation quality.
- Knowledge distillation methods, such as Ensemble KD and R2L KD, are effective in Chinese-English translation.

As a result, we use all data augmentation methods described above to train our final models. After ensemble 5 deep models, we obtain 11.01 BLEU improvements over the baseline system. Our final submission with TEL achieves 48.12 BLEU-SBP on the ccmt2020 test set, which gains 1.1 BLEU-SBP improvements over the submission with standard model ensemble.

3.3 Results on English-Chinese Translation

Table 3 shows the results of English-Chinese Translation on newstest2019 test set. Due to limited resources, we only use the back-translation method to augment data in this task. As we can see from the table, the results of BT and noise BT are nearly identical, which do not coincides with the findings in Chinese-English translation. Furthermore, we do not found the benefits of deep models on this task. Finetuning with in-domain data brings substantial improvements, but not as effective as Chinese-English translation. After ensembling 4 models finetuned with in-domain data, we finally obtain 39.56 BLEU on newstest2019. Our final submission achieves 63.43 BLEU-SBP on ccmt2020 English-Chinese test set.

4 Conclusion

This paper described the neural machine translation systems developed by Tsinghua University in the CCMT 2020 bilingual translation tasks. We exploited deep models, various data augmentation methods, finetuning techniques, as well as model ensembles in our experiments. We verified through experiments that combining all these methods can lead to substantial improvements in translation quality.

Acknowledgements

This work was supported by the National Key R&D Program of China (No. 2017YFB0 202204), National Natural Science Foundation of China (No. 61925601, No. 61761166 008, No. 61772302), Beijing Academy of Artificial Intelligence, and the NExT++ project supported by the National Research Foundation, Prime Ministers Office, Singapore under its IRC@Singapore Funding Initiative.

References

1. Edunov, S., Ott, M., Auli, M., Grangier, D.: Understanding back-translation at scale. In: Proceedings of EMNLP (2018)
2. Kim, Y., Rush, A.M.: Sequence-level knowledge distillation. In: Proceedings EMNLP (2016)
3. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: Proceedings of ICLR (2015)
4. Sennrich, R., Haddow, B., Birch, A.: Improving neural machine translation models with monolingual data. In: Proceedings of ACL (2016)
5. Sennrich, R., Haddow, B., Birch, A.: Neural machine translation of rare words with subword units. In: Proceedings of ACL (2016)
6. Sun, M., Jiang, B., Xiong, H., He, Z., Wu, H., Wang, H.: Baidu neural machine translation systems for wmt19. In: Proceedings of the Fourth Conference on Machine Translation (2019)
7. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Proceedings of NeurIPS (2017)
8. Wang, Q., Li, B., Xiao, T., Zhu, J., Li, C., Wong, D.F., Chao, L.S.: Learning deep transformer models for machine translation. In: Proceedings of ACL (2019)
9. Wang, Y., Wu, L., Xia, Y., Qin, T., Zhai, C., Liu, T.Y.: Transductive ensemble learning for neural machine translation. In: AAAI (2020)
10. Wei, H.R., Huang, S., Wang, R., Dai, X.y., Chen, J.: Online distilling from checkpoints for neural machine translation. In: Proceedings of NAACL-HLT (2019)