IIE's Neural Machine Translation Systems for WMT20

Xiangpeng Wei[1,2], **Ping Guo**[1,2], **Yunpeng Li**[1,2], **Xingsheng Zhang**[1,2], **Luxi Xing**[1,2], **Yue Hu**[1,2]
[1]Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[2]School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
{weixiangpeng,huyue}@iie.ac.cn

Abstract

In this paper we introduce the systems IIE submitted for the WMT20 shared task on German↔French news translation. Our systems are based on the Transformer architecture with some effective improvements. Multiscale collaborative deep architecture, data selection, back translation, knowledge distillation, domain adaptation, model ensemble and re-ranking are employed and proven effective in our experiments. Our German→French system achieved 35.0 BLEU and ranked the second among all anonymous submissions, and our French→German system achieved 36.6 BLEU and ranked the fourth in all anonymous submissions.

1 Introduction

We participate in the WMT20 shared news translation task in one language pair and two language directions, German→French and French→German. Our methods are based on techniques and approaches used in submissions from past years (Deng et al., 2018; Ng et al., 2019; Sun et al., 2019; Li et al., 2019; Xia et al., 2019), including the use of subword models (Sennrich et al., 2016), iterative back-translation, knowledge distillation, model ensembling and several techniques we proposed recently (Wei et al., 2020b,a).

For our submissions of two language directions, we adopt the deep transformer architectures (48-layer) based on multiscale collaboration mechanism (Wei et al., 2020b) as our baseline, which outperformed the standard `Transformer-Big` as well as shallower models significantly in terms of translation quality. We also use an iterative back-translation approach (Zhang et al., 2018) with the controllable sampling to extend the back translation method by jointly training source-to-target and target-to-source NMT models. Moreover, the knowledge distillation (Freitag et al., 2017) is employed to leverage the source-side monolingual data. For our final models, we apply a domain-specific fine-tuning process and model ensembling, and decode using noisy channel model re-ranking.

The paper is structured as follows: Section 2 describes the techniques we used, then section 3 shows the experimental settings and results. Finally, we conclude our work in Section 4.

2 Our Techniques

2.1 Multiscale Collaborative Deep Models

The structure of NMT models has evolved quickly, such as RNN-based (Wu et al., 2016), CNN-based (Gehring et al., 2017) and attention-based (Vaswani et al., 2017) systems. Deep neural networks have revolutionized the state-of-the-art in various communities, from computer vision to natural language processing. We adopt the deep transformer model proposed by our work (Wei et al., 2020b). Instead of relying on the whole encoder stack to directly learn a desired representation, we let each encoder block learn a fine-grained representation and enhance it by encoding spatial dependencies using a bottom-up network. For coordination, we attend each block of the decoder to both the corresponding representation of the encoder and the contextual representation with spatial dependencies. This not only shortens the path of error propagation, but also helps to prevent the lower level information from being forgotten or diluted. In this section we describe the details (as illustrated in figure 1) of our deep architectures as below:

Block-Scale Collaboration. An intuitive extension of naive stacking of layers is to group few stacked layers into a *block*. We suppose that the encoder and decoder of our model have the same number of blocks (i.e., N). Each block of the encoder has M_n ($n \in \{1, 2, ..., N\}$) identical layers,

Proceedings of the 5th Conference on Machine Translation (WMT), pages 300–304
Online, November 19–20, 2020.

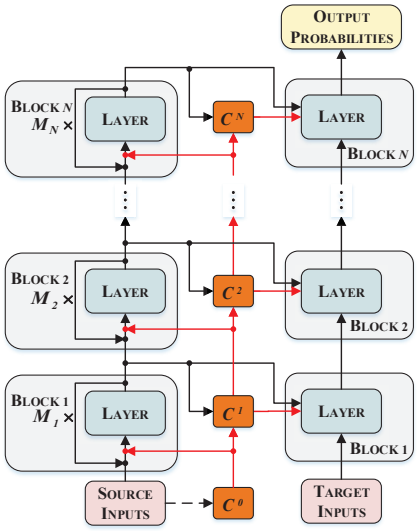

Figure 1: Illustration of Multiscale Collaborative Deep NMT Model. N is the number of encoder and decoder blocks. The n-th block of the encoder consists of M_n layers, while each decoder block only contains one layer.

while each decoder block contains one layer. Thus, we can adjust the value of each M_n flexibly to increase the depth of the encoder. Formally, for the n-th block of the encoder:

$$\mathrm{B}_e^n = \textsc{Block}_e(\mathrm{B}_e^{n-1}), \tag{1}$$

where $\textsc{Block}_e(\cdot)$ is the block function, in which the layer function $\mathcal{F}(\cdot)$ is iterated M_n times, i.e.

$$\begin{aligned} \mathrm{B}_e^n &= \mathrm{H}_e^{n,M_n}, \\ \mathrm{H}_e^{n,l} &= \mathcal{F}(\mathrm{H}_e^{n,l-1}; \mathbf{\Theta}_e^{n,l}) + \mathrm{H}_e^{n,l-1}, \\ \mathrm{H}_e^{n,0} &= \mathrm{B}_e^{n-1}, \end{aligned} \tag{2}$$

where $l \in \{1, 2, ..., M_n\}$, $\mathrm{H}_e^{n,l}$ and $\mathbf{\Theta}_e^{n,l}$ are the representation and parameters of the l-th layer in the n-th block, respectively. The decoder works in a similar way but the layer function $\mathcal{G}(\cdot)$ is iterated only once in each block,

$$\begin{aligned} \mathrm{B}_d^n &= \textsc{Block}_d(\mathrm{B}_d^{n-1}, \mathrm{B}_e^n) \\ &= \mathcal{G}(\mathrm{B}_d^{n-1}, \mathrm{B}_e^n; \mathbf{\Theta}_d^n) + \mathrm{B}_d^{n-1}. \end{aligned} \tag{3}$$

Each block of the decoder attends to the corresponding encoder block.

Contextual Collaboration. To model long-term spatial dependencies and reuse global representations, we define a GRU cell $\mathcal{Q}(\mathbf{c}, \bar{\mathbf{x}})$, which maps a hidden state $\mathbf{c}$ and an additional input $\bar{\mathbf{x}}$ into a new hidden state:

$$\begin{aligned} \mathrm{C}^n &= \mathcal{Q}(\mathrm{C}^{n-1}, \mathrm{B}_e^n), n \in [1, N] \\ \mathrm{C}^0 &= \mathcal{E}_e, \end{aligned} \tag{4}$$

where $\mathcal{E}_e$ is the embedding matrix of the source input $\mathbf{x}$. The new state C^n can be fused with each layer of the subsequent blocks in both the encoder and the decoder. Formally, B_e^n in Eq.(1) can be re-calculated in the following way:

$$\begin{aligned} \mathrm{B}_e^n &= \mathrm{H}_e^{n,M_n}, \\ \mathrm{H}_e^{n,l} &= \mathcal{F}(\mathrm{H}_e^{n,l-1}, \mathrm{C}^{n-1}; \mathbf{\Theta}_e^{n,l}) + \mathrm{H}_e^{n,l-1}, \\ \mathrm{H}_e^{n,0} &= \mathrm{B}_e^{n-1}. \end{aligned} \tag{5}$$

Similarly, for decoder, we have

$$\begin{aligned} \mathrm{B}_d^n &= \textsc{Block}_d(\mathrm{B}_d^{n-1}, \mathrm{B}_e^n) \\ &= \mathcal{G}(\mathrm{B}_d^{n-1}, \mathrm{B}_e^n, \mathrm{C}^n; \mathbf{\Theta}_d^n) + \mathrm{B}_d^{n-1}. \end{aligned} \tag{6}$$

2.2 Back-Translation with Controllable Sampling

Back-translation (BT) is an effective and commonly used data augmentation technique to incorporate monolingual data into a translation system. Back-translation first trains an intermediate target-to-source system that is used to translate monolingual target data into additional synthetic parallel data. This data is used in conjunction with human translated bitext data to train the desired source-to-target system.

In our work, we use an iterative back-translation approach to jointly train source-to-target and target-to-source NMT models. The process can be summarized as below:

- step 1: we train both a source-to-target model ($\mathcal{M}^0_{x\to y}$) and a target-to-source model ($\mathcal{M}^0_{y\to x}$) using the human translated data.
- step 2: we use $\mathcal{M}^t_{x\to y}$ to translate source-side monolingual data to target language, and use $\mathcal{M}^t_{y\to x}$ to translate target-side monolingual data to source language, where t starts from 0.
- step 3: we combine both the human translated data and pseudo data synthesized in step 2 to further optimize the two NMT models respectively.
- Repeat steps 2-3 until the models converge.

In practice, we repeat 3 times for steps 2-3. We apply the controllable sampling strategy (Wei et al., 2020a) to synthesize reasonable sentences which are at both high quality and diversity.

2.3 Knowledge Distillation and Ensemble

The early adoption of knowledge distillation (KD) (Kim and Rush, 2016) is for model compression. We use the same method as in Sun et al. (2019) that adopts hybrid heterogeneous teacher: base transformer, deep transformer, big transformer and RNMT+ (Chen et al., 2018). For each individual model, we use the other two models as the teacher model to further improve the performance. In addition, model ensemble is also used to boost the performance by combining the predictions of above four models at each decoding step.

2.4 Domain-specific Fine-tuning

Fine-tuning with domain-specific data is a common and effective method to improve translation quality for a downstream task. After completing training on the bitext and back-translated data, we train for an additional epoch on a smaller in-domain corpus. We first select 100K sentence-pairs from the bilingual as well as pseudo-generated data according to the filter method in Deng et al. (2018) and continue to train the model on the filtered data.

2.5 Reranking

N-best reranking is a method of improving translation quality by scoring and selecting a candidate hypothesis from a list of n-best hypotheses generated by a source-to-target model. For our submissions, we rerank the n-best hypotheses using two aspects as follows:

$$\log p(y|x) + \lambda_1 \log p(x|y) + \lambda_2 \log p(y) \quad (7)$$

The weights λ_1 and λ_2 are determined by tuning them with a random search on a validation set and selecting the weights that give the best performance.

3 System Overview

We submit constrained systems to both German to French and French to German translations, with the same techniques.

3.1 Dataset

We use all available bilingual datasets and select 10M bilingual data from WMT'20 corpora using the script `filter_interactive.py`[1]. We share a vocabulary for the two languages and apply BPE for word segmentation with 32K merge operations. For monolingual data, we use 18M German sentences and 18M French sentences from Newscrawl, and pre-process them in the same way as bilingual data. We split 9k sentences from the "dev08-14" as the validation set and use newstest 2019 as the test set.

[1]Scripts at: https://tinyurl.com/yx9fpoam.

System	German→French	
	Dev	Newstest19
MSC (48L)	28.9	33.2
+ Iterative BT	31.2	35.7
+ KD & Ensemble	32.3	36.5
+ Fine-tuning	32.9	37.2
+ Reranking	33.8	38.4
WMT'20 submission		**35.0**

Table 1: SacreBLEU scores on German→French.

System	French→German	
	Dev	Newstest19
MSC (48L)	22.8	31.7
+ Iterative BT	24.2	34.0
+ KD & Ensemble	25.1	34.7
+ Fine-tuning	25.9	35.4
+ Reranking	26.5	36.3
WMT'20 submission		**36.6**

Table 2: SacreBLEU scores on French→German.

3.2 Model Configuration

We use the PyTorch implementation of Transformer[2]. We choose the `Transformer_base` setting, in which the encoder and decoder are of 48 and 6 layers, respectively. The dropout rate is fixed as 0.1. We set the batch size as 4096 and the parameter `--update-freq` as 16.

3.3 Results

Results and ablations for De→Fr Fr→De are shown in Table 1 and 2, respectively. We report case-sensitive SacreBLEU scores using SacreBLEU (Post, 2018)[3], using international tokenization for German↔French.

German→French For De→Fr, iterative BT improves our baseline performance on newstest 2019

[2]https://github.com/pytorch/fairseq

[3]SacreBLEU signatures: BLEU+case.mixed+lang.de-fr+numrefs.1+smooth.exp+test.wmt19+tok.13a+version.1.2.11, BLEU+case.mixed+lang.fr-de+numrefs.1+smooth.exp+test.wmt19+tok.13a+version.1.2.11

by about 2.5 BLEU. The addition of KD and model ensemble improves single model performance by 0.8 BLEU, but combining this with fine-tuning and reranking gives us a total of 2 BLEU. Our final submission for WMT20 achieves 35.0 BLEU points for German→French translation (ranked in the second place).

French→German For Fr→De, we see similar improvements with iterative BT by about 2.3 BLEU. KD, ensembling, and fine-tuning add an additional 1.4 BLEU, with reranking contributing 0.9 BLEU. Our final submission for WMT20 achieves 36.6 BLEU points for French→German translation (ranked in the fourth among anonymous submissions).

4 Conclusion

This paper describes CAS IIE's submission to the WMT20 German↔French news translation task. We investigate extremely deep models (with 48 layers) and exploit effective strategies to better utilize parallel data as well as monolingual data. Finally, our German→French system achieved 35.0 BLEU and ranked the second among all anonymous submissions, and our French→German system achieved 36.6 BLEU and ranked the fourth in all anonymous submissions.

References

Mia Xu Chen, Orhan Firat, Ankur Bapna, Melvin Johnson, Wolfgang Macherey, George Foster, Llion Jones, Mike Schuster, Noam Shazeer, Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Zhifeng Chen, Yonghui Wu, and Macduff Hughes. 2018. The best of both worlds: Combining recent advances in neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 76–86, Melbourne, Australia. Association for Computational Linguistics.

Yongchao Deng, Shanbo Cheng, Jun Lu, Kai Song, Jingang Wang, Shenglan Wu, Liang Yao, Guchun Zhang, Haibo Zhang, Pei Zhang, Changfeng Zhu, and Boxing Chen. 2018. Alibaba's neural machine translation systems for wmt18. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 372–380, Belgium, Brussels. Association for Computational Linguistics.

Markus Freitag, Yaser Al-Onaizan, and Baskaran Sankaran. 2017. Ensemble distillation for neural machine translation. *CoRR*, abs/1702.01802.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. In *arXiv:1705.03122*.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327, Austin, Texas. Association for Computational Linguistics.

Bei Li, Yinqiao Li, Chen Xu, Ye Lin, Jiqiang Liu, Hui Liu, Ziyang Wang, Yuhao Zhang, Nuo Xu, Zeyang Wang, Kai Feng, Hexuan Chen, Tengbo Liu, Yanyang Li, Qiang Wang, Tong Xiao, and Jingbo Zhu. 2019. The niutrans machine translation systems for wmt19. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 257–266, Florence, Italy. Association for Computational Linguistics.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook fair's wmt19 news translation task submission. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 314–319, Florence, Italy. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. 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.

Meng Sun, Bojian Jiang, Hao Xiong, Zhongjun He, Hua Wu, and Haifeng Wang. 2019. Baidu neural machine translation systems for wmt19. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 374–381, Florence, Italy. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 5998–6008.

Xiangpeng Wei, Heng Yu, Yue Hu, Rongxiang Weng, Luxi Xing, and Weihua Luo. 2020a. Uncertainty-aware semantic augmentation for neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020*. Association for Computational Linguistics.

Xiangpeng Wei, Heng Yu, Yue Hu, Yue Zhang, Rongxiang Weng, and Weihua Luo. 2020b. Multiscale collaborative deep models for neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 414–426, Online. Association for Computational Linguistics.

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

Yingce Xia, Xu Tan, Fei Tian, Fei Gao, Di He, Weicong Chen, Yang Fan, Linyuan Gong, Yichong Leng, Renqian Luo, Yiren Wang, Lijun Wu, Jinhua Zhu, Tao Qin, and Tie-Yan Liu. 2019. Microsoft research asia's systems for wmt19. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 424–433, Florence, Italy. Association for Computational Linguistics.

Zhirui Zhang, Shujie Liu, Mu Li, Ming Zhou, and Enhong Chen. 2018. Joint training for neural machine translation models with monolingual data. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 555–562. AAAI Press.