Tencent Neural Machine Translation Systems for the WMT20 News Translation Task

Shuangzhi Wu* SPPD of Tencent
Xing Wang* Tencent AI Lab
Longyue Wang* Tencent AI Lab
Fangxu Liu* SPPD of Tencent
Jun Xie SPPD of Tencent
Zhaopeng Tu Tencent AI Lab
Shuming Shi Tencent AI Lab
Mu Li SPPD of Tencent

Abstract

This paper describes Tencent Neural Machine Translation systems for the WMT 2020 news translation tasks. We participate in the shared news translation task on English ↔ Chinese and English → German language pairs. Our systems are built on deep Transformer and several data augmentation methods. We propose a boosted in-domain finetuning method to improve single models. Ensemble is used to combine single models and we propose an iterative transductive ensemble method which can further improve the translation performance based on the ensemble results. We achieve a BLEU score of 36.8 and the highest chrF score of 0.648 on Chinese → English task.

1 Introduction

Recently, Transformer (Vaswani et al., 2017), that depends on self-attention mechanism , has significantly improved the translation quality. It is widely used as basic Neural Machine Translation (NMT) models in previous WMT translation tasks (Wang et al., 2018b; Li et al., 2019; Sun et al., 2019). In this year's translation task, our Tencent Translation team participated in three WMT2020 shared news translation tasks, including Chinese → English, English → Chinese and English → German. For the three tasks, we use similar model architectures and training strategies. Four structures are used and all of them are based on deep transformer which are proven more effective than the standard Transformer-big models (Li et al., 2019).

In terms of data augmentation, we adopt R2L training (Zhang et al., 2019) to all the tasks. Monolingual data is only used in English → German task following the back-translation manner (Sennrich et al., 2016b). Different from the standard back-translation, we add noise to the synthetic source sentence in order to take advantage of large-scale monolingual text. In addition, we add a special token to the synthetic source sentence to help the model better distinguish the bilingual data and synthetic data. The in-domain finetuning (Sun et al., 2019) is very effective in our three experiments and specially, we propose a boosted finetuning method for English ↔ Chinese tasks. We also take advantage of the combination methods to further improve the translation quality. The "greedy search ensemble algorithm" (Li et al., 2019) is used to select the best combinations from single models. Then for English ↔ Chinese tasks we propose an iterative transductive ensemble (ITE) method based on the translation results of the ensemble models. For English → German task, we apply the noise channel model for re-ranking (Yee et al., 2019).

This paper was structured as follows: Section 2 describes the dataset. We present the detailed overview of our system in Section 3. The experiment settings and main results are shown in Section 4. Finally, we conclude our work in Section 5.

2 Dataset

2.1 Chinese ↔ English

The bilingual data used in Chinese ↔ English task includes all the available corpus provided by WMT2020: News Commentary v15, Wiki Titles v2, UN Parallel Corpus V1.0, CCMT Corpus, WikiMatrix, Back-translated news. The Chinese sentences are segmented by jieba segmentor[1] while the English side is processed by Moses tokenizer. We collect 18M sentence pairs after filtering.

2.2 English → German

The bilingual data used in this task includes all the available corpus provided by WMT2020. For the Paracrawl part, We filter most of the data due to

* Equal contribution. Correspondence to {*frostwu*, *brightxwang*, *vinnylywang*, *fangxuliu*}@*tencent.com*.

[1] https://github.com/fxsjy/jieba.

Proceedings of the 5th Conference on Machine Translation (WMT), pages 313–319
Online, November 19–20, 2020. ©2020 Association for Computational Linguistics

bad quality and collect 15M sentence pairs. Totally, 22M sentence pairs are used for training. Both the languages are tokenized by *tokenize.perl* script[2]. Then BPE is applied with 32K operations. The vocabulary is shared with 32K unique words. For monolingual data, we randomly select 80M sentences from NewsCrawl2017-2019 for back-translation and 45M are used for training after filtering

2.3 Data Processing

Pre-processing To pre-process the raw data, we apply a series of open-source/in-house scripts, including full-/half-width conversion, Unicode conversation, punctuation normalization, tokenization and true-casing. After filtering steps, we generated subwords via BPE (Sennrich et al., 2016c) with pre-defined merge operations of 32,000.

Filtering To improve the quality of data, we filtered noisy sentence pairs according to their characteristics in terms of language identification, duplication, length, invalid string and edit distance. More specifically, we filter out the sentences longer than 150 words. The word ratio between the source and the target must not exceed 1:1.3 or 1.3:1. According to our observations, the filtering method can significantly reduce noise issues including misalignment, translation error, illegal characters, over-translation and under-translation.

3 System Overview

3.1 Model Architecture

In our systems, we adopt four different model architectures with TRANSFORMER (Vaswani et al., 2017):

- **DEEP** TRANSFORMER (Dou et al., 2018; Wang et al., 2019; Dou et al., 2019) is the TRANSFORMER-BASE model with the 40-layer encoder.

- **HYBRID** TRANSFORMER (Hao et al., 2019) is the TRANSFORMER-BASE model with 40-layer hybrid encoder. The 40-layer hybrid encoder stacks 35-layer self-attention-based encoder on top of 5-layer bi-directional ON-LSTM (Shen et al., 2019) encoder.

- **BIGDEEP** TRANSFORMER is the TRANSFORMER-BIG model with 20 encoder layers.

- **LARGER** TRANSFORMER is similar to BIGDEEP model except that it uses 8192 as the FFN inner width.

The main differences between these models are presented in Table 1. To stabilize the training of deep model, we use the Pre-Norm strategy (Li et al., 2019). The layer normalization was applied to the input of every sub-layer which the computation sequence could be expressed as: normalize → Transform → dropout → residual-add. All models are implemented on top of the open-source toolkit Fairseq[3] (Ott et al., 2019).

3.2 Data Augmentation

Data augmentation is a commonly used technique to improve the translation quality. There are various of methods to conduct data augmentation such as back-translation (Sennrich et al., 2016a), joint training (Zhang et al., 2018) etc. In this section, we will introduce the methods we used in WMT2020.

3.2.1 Large-scale Back-translation

Back-translation is the most commonly used data augmentation technique to incorporate monolingual data into NMT (Sennrich et al., 2016a). The method first trains an intermediate target-to-source system, which is used to translate target monolingual corpus into source. Then the synthetic parallel corpus is used to train models together with the bilingual data.

In this work we apply the noise back-translations method as introduced in (Lample et al., 2018). When translating monolingual data we use an ensemble of two models to get better source translations. We follow (Edunov et al., 2018) to add noise to the synthetic source data. Furthermore, we use a tag at the head of each synthetic source sentence as Caswell et al. (2019) does. To filter the pseudo corpus, we translate the synthetic source into target and calculate a Round-Trip BLEU score, the synthetic pairs are dropped if the BLEU score is lower than 30. Notably, we only apply back translation to the English → German task. We find that back translation decrease the translation quality to Chinese ↔ English tasks in our experiments.

[2] https://github.com/moses-smt/mosesdecoder/tree/master/scripts/tokenizer/tokenizer.perl

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

	DEEP	HYBRID	BIGDEEP	LARGER
Encoder Layer	40	40	20	20
Decoder Layer	6	6	6	6
Attention Heads	8	8	16	16
Embedding Size	512	512	1024	1024
FFN Size	2048	2048	4096	8192

Table 1: Hyper-parameters of different Transformer models used in our system.

3.2.2 R2L Training

The approach is proposed by (Zhang et al., 2019). The main idea is to integrate the information of Right-to-Left (R2L) models to Left-to-Right (L2R) ones. Following this work, we translate the source sentences of the parallel data with both a R2L model and a L2R model, and use the translated pseudo corpus to improve the L2R model. We drop the pseudo parallel data if the BLEU score lower than 15. This method is applied to all the three tasks.

3.3 Finetuning

We use in-domain finetuning to further improve the model performance on news domain as previous study (Sun et al., 2019) shows that finetuning is very effective on the WMT2019 news translation tasks. For the three tasks, the finetuning is slight different and we will introduce them seprately in the following of this section.

Finetuning Zh → En Models For this task, we use all the previous development and test dataset as in-domain corpus D that includes WMT2017 development data, WMT2017 test data and WMT2018 test data. After training an NMT model M with the above methods, we finetune W on D with the same hyper parameters of training M. When testing on the WMT2019 test set, we achieve about 4-5 BLEUs improvement. As the in-domain corpus is very limited, we propose a boosted finetuning method by using the R2L training method to boost the finetuning process, which is named finetuning (boost). In our final submission, we add the WMT2019 test to D, the batch size is set to 2,048, the finetuning finished after 3k training steps.

Finetuning En → Zh Models We select the WMT2017 development data, WMT2017 test data and WMT2018 test data as the in domain corpus D in both tuning models and final submission which is different from Zh → En task. In addition, we do not use R2L training or add WMT2019 test to D, as we find this is useless. When finetuning, we reset the optimizer and use a fixed learning rate of 8e-5. The batch size is set to 1024 and the finetuning finishes after 900 upates.

Finetuning En → De Models We select the document whose source side is originally in English from all previous development and test dataset as in-domain corpus D. Single models are trained with the above methods are then finetune on D for one epoch with a fixed learning rate of 1e-4. In our final submission, the WMT2019 test set is added to D for better performance improvement.

3.4 Re-ranking

We use noisy channel model re-ranking method (Yee et al., 2019). This method is implemented in Fairseq [4]. Three features are used as following:

Source-to-Target Model Instead of a single model, we use the ensemble model as source-to-target model. Four well-trained single models are used. The decoding beam size is set to 25. We collect the log probability of each translation candidates.

Target-to-Source Model The target-to-source model is the channel mode which is used to translate the candidates back to source. We use a big transformer model for target-to-source.

Language Models For language model, we train a small GPT-2 model with FFN=8192 for target monolingual data.

Tuning We use random search to choose values in the range $[0, 3)$ for λ_1 , λ_2 and length penalty. The parameters are tuned on development set.

3.5 Ensemble

Model ensemble is a widely used technique in previous WMT workshops (Li et al., 2019; Sun et al.,

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

2019; Wang et al., 2018a) which can boost the performance by combining the predictions of several models at each decoding step. In our work, we use two kinds of ensemble methods and finally the two are combined for further improvements.

3.6 Greedy Based Ensemble

This method is proposed by Li et al. (2019). The method adopts an easy operable greedy-base strategy to search for a better single model combinations on the development set. For more detail, please refer to the original paper. We also train single models with different hyper parameters to ensure the diversity. We refer to this method as **Ensemble** in the following.

3.7 Iterative Transductive Ensemble

Transductive ensemble (TE) is proposed by Wang et al. (2020c). The key idea is that source input sentences from the validation and test sets are firstly translated to the target language space with multiple different well-trained NMT models, which results in a pretranslated synthetic dataset. Then individual models are finetuned on the generated synthetic dataset. We propose an variation of TE, the Iterative Transductive Ensemble (ITE) which is based on Ensemble, as following:

Algorithm 1: Iterative Transductive Ensemble

```
Input: Single models $M_1^m$, In-domain corpus $D$, $E_1^n$
       is $n$ different ensemble combinations
Output: Single models $M_1^m$
Translate $D$ with $E_1^n$ and get ${D'_1}^n$
Train each $M_1^m$ on $D \cup {D'_1}^n$ and get ${M'_1}^m$, then
  $M_1^m = {M'_1}^m$
$t := 0$
while not convergence do
    Translate $D$ with $M_1^m$ and get ${D''_1}^n$, then
      ${D'_1}^n = {D'_1}^n \cup {D''_1}^n$
    Train each $M_1^m$ on $D \cup {D'_1}^n$ and get ${M'_1}^m$, then
      $M_1^m = {M'_1}^m$
    $t := t + 1$
return ,
```

4 Experiments and Results

4.1 Setups

The implementation of our models is based on Fairseq [5]. All the single models are carried out on 8 NVIDIA V100 GPUs each of which have 32 GB memory. We use the Adam optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.98$. The gradient accumulation is used due to the high GPU memory consumption. The batch size is set to 8192 toknes per GPU and the "update-freq" parameter in Fairseq is set to 8. Specifically, for LARGE settings, the batch size is 4096 and "update-freq" is 16. We set max learning rate to 0.0007 and warmup-steps to 4000. All the dropout probabilities are set to 0.1. We select the checkpoint with the lowest loss on development set as the final checkpoint in each training. We calculate sacreBLEU score [6] for all experiments which is officially recommended. The WMT2019 testset (test2019) is used as the development set for all the tasks.

4.2 Chinese → English

Table 2 shows the Chinese → English translation results on validation set. We train multiple single models in each settings and report the best scores in Table 2. The R2L method can significantly improve the baseline by 2.45 BLEU scores. It is surprising to find a gain of almost 5 BLEU improvement on test2019 dataset. After we boost the in-domain corpus, we can achieve 1 more BLEU on the DEEP model. This illustrates that the finetuning is very effective on the WMT2019 test set.

In our experiments, the ensemble models consists of 5 single models: 1 HYBRID, 1 BIGDEEP, 3 LARGER models. As shown in the Table2, the ensemble models outperform the best single model by 1.06 BLEU score. We then apply transductive ensemble to LARGER models and finally the performance achieves 38.99. We also find that the single models that applied TE cannot bring further improvement to ensemble results. We do not apply re-ranking to this task, as we find that the improvement is insignificant. Our WMT 2020 Chinese → English submission achieves a SacreBLEU score of 36.8 and chrF score of 0.649.

4.3 English → Chinese

Table 3 shows the English → Chinese translation results on validation set. We also train multiple single models and report the best scores in the Table. After applying R2L method, we achieve 0.4 to 1 BLEU. We can observe that the improvement from finetuning is not as high as Chinese → English tasks, where only more 1 BLEU is gained. We also find that the boosted finetuning is harmful in this task, thus we omit the results. The ensemble

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

[6] https://github.com/mjpost/sacrebleu

	Deep	Hybrid	BigDeep	Larger
Baseline	29.01	-	-	-
+R2L	31.46	31.42	32.07	32.41
+Finetuning	36.04	-	-	-
+Finetuning(boost)	37.02	37.23	37.38	37.62
Ensemble	38.68			
ITE	38.99			

Table 2: BLEU evaluation results on the WMT 2019 Chinese → English test set.

	Deep	BigDeep	Larger
Baseline	38.10	38.63	38.90
+R2L	39.09	39.01	39.31
+Finetuning	-	40.72	40.68
Ensemble	41.46		
ITE	42.26		

Table 3: BLEU evaluation results on the WMT 2019 English → Chinese test set.

	BigDeep	Deep
Baseline	41.58	41.71
+R2L	43.05	42.73
+BT	44.37	44.06
+Finetuning	45.30	44.82
+Ensemble	45.7	
+reranking	45.9	
+PostProcessing	47.3	

Table 4: BLEU evaluation results on the WMT 2019 English → German test set.

setting consist of 4 models, that are 2 BigDeep, 2 Larger models, which outperform the best single model by 0.74 BLEU.

4.4 English → German

Table 4 shows the results on English → German translation. The baseline is the BigDeep model using only bilingual data. R2L training boosts the BLEU score from 41.58 to 43.05. After adding back-translation, we further improve the BLEU score to 44.37. The finetuning can further achieve 0.93 BLEU improvement on the BigDeep model.

In this task, the ensemble models consists of 4 single models: 1 Deep, 2 BigDeep, 1 Larger models. As shown in Table 4, the ensemble models outperform the best single model by 0.4 BLEU score. We then apply noisy channel re-reranking to ensemble results and finally achieve 45.9 BLEU on the development set.

We apply a post-processing procedure. After translating the source-side, we normalize the English quotations appearing in the German translations to German-style quotations. We find this can improve the BLEU score on development set by 1.4 points.

5 Conclusion

This paper presents the Tencent Translation systems for WMT2020 Chinese → English news translation tasks. We investigate various deep architectures to build strong baseline systems. Then popular data augmentation methods such as back-translation and R2L training are used to improve the baselines. We also prove that in-domain finetuning is very effective for news translation tasks especially on Chinese → English task. Finally, we adopt the greed-based ensemble algorithm and propose an iterative transductive ensemble method for further improvement.

It is worth mentioning a number of advanced technologies reported in this paper are also adapted to our systems for biomedical translation (Wang et al., 2020b) and chat translation (Wang et al., 2020a) tasks, which respectively achieve up to 1st and 2nd ranks in terms of BLEU scores.

References

Isaac Caswell, Ciprian Chelba, and David Grangier. 2019. Tagged back-translation. *CoRR*, abs/1906.06442.

Zi-Yi Dou, Zhaopeng Tu, Xing Wang, Shuming Shi, and Tong Zhang. 2018. Exploiting deep representations for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4253–4262, Brussels, Belgium. Association for Computational Linguistics.

Zi-Yi Dou, Xing Wang, Shuming Shi, and Zhaopeng Tu. 2019. Exploiting deep representations for natural language processing. *Neurocomputing*.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500, Brussels, Belgium. Association for Computational Linguistics.

Jie Hao, Xing Wang, Shuming Shi, Jinfeng Zhang, and Zhaopeng Tu. 2019. Towards better modeling hierarchical structure for self-attention with ordered neurons. In *EMNLP-IJCNLP*, pages 1336–1341.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Unsupervised machine translation using monolingual corpora only. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*.

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.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. Fairseq: A fast, extensible toolkit for sequence modeling. *NAACL*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. 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.

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

Yikang Shen, Shawn Tan, Alessandro Sordoni, and Aaron Courville. 2019. Ordered neurons: Integrating tree structures into recurrent neural networks. In *ICLR*.

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, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*.

Longyue Wang, Zhaopeng Tu, Wang Xing, Li Ding, Liang Ding, and Shuming Shi. 2020a. Tencent AI Lab machine translation systems for the WMT20 chat translation task. In *Proceedings of the Fifth Conference on Machine Translation*.

Mingxuan Wang, Li Gong, Wenhuan Zhu, Jun Xie, and Chao Bian. 2018a. Tencent neural machine translation systems for wmt18. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 526–531, Belgium, Brussels. Association for Computational Linguistics.

Qiang Wang, Bei Li, Jiqiang Liu, Bojian Jiang, Zheyang Zhang, Yinqiao Li, Ye Lin, Tong Xiao, and Jingbo Zhu. 2018b. The niutrans machine translation system for wmt18. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 532–538, Belgium, Brussels. Association for Computational Linguistics.

Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F Wong, and Lidia S Chao. 2019. Learning deep transformer models for machine translation. In *ACL*.

Xing Wang, Zhaopeng Tu, Longyue Wang Wang, and Shuming Shi. 2020b. Tencent AI Lab machine translation systems for the WMT20 biomedical translation task. In *Proceedings of the Fifth Conference on Machine Translation*.

Yiren Wang, Lijun Wu, Yingce Xia, Tao Qin, ChengXiang Zhai, and Tie-Yan Liu. 2020c. Transductive ensemble learning for neural machine translation. In *AAAI*, pages 6291–6298.

Kyra Yee, Yann Dauphin, and Michael Auli. 2019. Simple and effective noisy channel modeling for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5696–5701, Hong Kong, China. 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. *arXiv preprint arXiv:1803.00353*.

Zhirui Zhang, Shuangzhi Wu, Shujie Liu, Mu Li, Ming Zhou, and Tong Xu. 2019. Regularizing neural machine translation by target-bidirectional agreement. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 443–450.