HW-TSC's Participation in the WMT 2021 Triangular MT Shared Task

Zongyao Li, Daimeng Wei, Hengchao Shang, Xiaoyu Chen,
Zhanglin Wu, Zhengzhe Yu, Jiaxin Guo, Minghan Wang,
Lizhi Lei, Min Zhang, Hao Yang, Ying Qin,
Huawei Translation Service Center, Beijing, China
{lizongyao,weidaimeng,shanghengchao,chenxiaoyu35,
wuzhanglin2,yuzhengzhe,guojiaxin1,wangminghan,
leilizhi,zhangmin186,yanghao30,qinying}@huawei.com

Abstract

This paper presents the submission of Huawei Translation Service Center (HW-TSC) to WMT 2021 Triangular MT Shared Task. We participate in the Russian-to-Chinese task under the constrained condition. We use Transformer architecture and obtain the best performance via a variant with larger parameter sizes. We perform detailed data pre-processing and filtering on the provided large-scale bilingual data. Several strategies are used to train our models, such as Multilingual Translation, Back Translation, Forward Translation, Data Denoising, Average Checkpoint, Ensemble, Fine-tuning, etc. Our system obtains 32.5 BLEU on the dev set and 27.7 BLEU on the test set, the highest score among all submissions.

1 Introduction

This paper introduces our submission to the WMT21 Triangular task. We adopt Transformer (Vaswani et al., 2017) architecture and strictly obey the constrained condition in terms of data usage. On one hand, we perform multiple data filtering strategies to enhance data quality; on the other hand, we leverage multilingual model (Johnson et al., 2017), pivot language, forward (Wu et al., 2019) and back translation (Edunov et al., 2018), and data denoising (Wang et al., 2018) strategies to further enhance training effects. In addition, we also adopt fine-tuning (Sun et al., 2019) and ensemble (Garmash and Monz, 2016), two widely used strategies, to further enhance system performance. We compare and contrast different strategies based on our experiment results and give our analysis accordingly.

The overall training process is illustrated in Figure 1. Section 2 mainly focuses on our training techniques, including model architecture, data processing and training strategies. Section 3 describes our experiment settings and training process. Section 4 presents the experiment results while section 5 analyze how our multilingual, data denoise and data augmentation strategies influence system performances.

2 Method

2.1 Model Architecture

Our system uses Transformer (Vaswani et al., 2017) model architecture, which adopts full self-attention mechanism to realize algorithm parallelism, accelerate model training speed, and improve translation quality. In this shared task, Transformer-Deep (Wang et al., 2019) is used, which features 35-layer encoder, 6-layer decoder, 768 dimensions of word vector, 3072-hidden-state, 16-head self-attention, and pre-norm.

2.2 Data Processing an Augmentation

We strictly comply with the constrained condition and use only the officially provided data.

2.2.1 Data Filtering

We perform the following steps to cleanse all data:

- Filter out repeated sentences (Khayrallah and Koehn, 2018; Ott et al., 2018).
- Convert XML escape characters.
- Normalize punctuations using Moses (Koehn et al., 2007).
- Delete html tags, non-UTF-8 characters, unicode characters and invisible characters.
- Filter out sentences with mismatched parentheses and quotation marks; sentences of which punctuation percentage exceeds 0.3; sentences with the character-to-word ratio greater than 12 or less than 1.5; sentences of which the source-to-target token ratio higher

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 325–330
November 10–11, 2021. ©2021 Association for Computational Linguistics

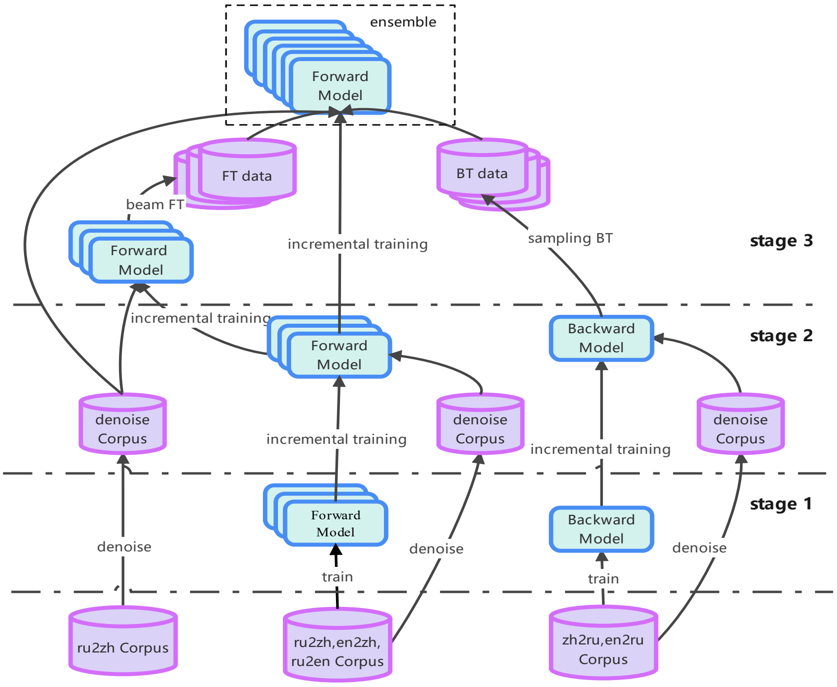

Figure 1: This figure shows the training process for the WMT 2021 Triangular MT Shared Task, which consists of three stages. In stage 1, three forward models and one backward model are trained. In stage 2, denoise corpus is used to train models incrementally. In stage 3, the synthetic data by FTST and denoise corpus are used to train models incrementally. Finally, model ensemble is used to boost the performance.

than 3 or lowers than 0.3; sentences with more than 120 tokens.

- Apply langid (Joulin et al., 2016b,a) to filter sentences in other languages.
- Use fast-align (Dyer et al., 2013) to filter sentence pairs with poor alignment, about 10% of the data is filtered.

We perform the additional steps to process Chinese data:

- Convert traditional Chinese characters to simplified ones.
- Convert fullwidth forms to halfwidth forms.

Data sizes before and after cleansing are listed in Table 1.

2.2.2 Data Augmentation

Back-translation (Edunov et al., 2018) is an effective way to boost translation quality by using monolingual data to generate synthetic training parallel data. As described in (Wu et al., 2019), similar to back translation, the monolingual corpus in source language can also be used to generate forward translation text with a trained MT model, and the generated forward and backward translation data can both be merged with the authentic bilingual data. This strategy can increase the data size to a large extent.

Since there is no officially provided monolingual data, we use the target side of en2zh data and the source side of zh2ru data filtered out in section 2.2.1 for back translation. We adopt the top-k sampling method. Then, we use the source side of ru2en data for forward translation, which is done based on beam search. Through sampling, we ensure that the sizes of data generated by forward and back translation are relatively equal. In this paper, we refer to the combination of forward and sampling back translation as FTST.

2.2.3 Filter Using LaBSE

Apart from the commonly used data cleansing methods, we also explore other techniques based on neural networks. LaBSE (Feng et al., 2020) is a multilingual BERT embedding model that can measure semantic similarities across languages. In our experiment, we notice that traditional data cleansing methods described in section 2.2.1 are unable to produce high-quality data, so we further filter the data using pre-training model LaBSE. For all parallel data, we calculated the similarity scores and

language pair	Raw data	Data Filtering	Filter Using LaBSE
en-zh	28.6M	14.7M	13.3M
en-ru	69.2M	45.1M	36.0M
ru-zh	33.4M	19.1M	14.7M

Table 1: Data sizes before and after filtering by different methods.

filtered out sentence pairs below a threshold. For Russian-Chinese data, the threshold is set to 0.7. For Russian-English and English-Chinese data, the threshold is set to 0.8. Our experiment integrates data denoising into the training process. The data size filtered by LaBSE is shown in table 1.

2.3 Multilingual Model

Johnson et al. (2017) proposes a simple solution that uses a single Neural Machine Translation (NMT) model to translate among multiple languages, and the model requires no change to the model architecture. Instead, the model introduces an artificial token at the beginning of the input sentence to specify the required target language. All languages use a shared vocabulary. There is no need to add more parameters. Surprisingly, experiments show that such model design can achieve better translation qualities across languages. In our experiment, we use two multilingual systems: forward model using ru2zh, en2zh, and ru2en data, and backward model using zh2ru and en2ru data.

2.4 Denoising Training

Wang et al. (2018) find that during training, dynamically adjusting noise data can boost system performance. The core idea is to train the model with noisy data at the initial stages and clearer data at later stages till the model converges. The quality of training data in this task is relatively poor as most of the data are crawled from website. We consider denoising training is suitable in this scenario. We simplify the denoising training process in our experiment, divide the training process into several stages.

For forward model, the training is divided into three steps: 1) Use all official provided data in three directions (ru2zh, en2zh, and ru2en) for training; 2) Use all clean data selected by LaBSE for incremental training; 3) Finally, use ru2zh clean data selected by LaBSE for incremental training.

For backward model, we only perform two steps: 1) Use all data (en2ru, zh2ru) for training; 2) Use zh2ru clean data selected by LaBSE for incremental training.

2.5 Fine-tuning and Ensemble

To achieve better results, fine-tuning with small-size in-domain data is necessary (Sun et al., 2019). An effective strategy for fine-tuning is to leverage the dev set available in this task. The fine-tuning strategies employed in our experiment include: 1) Add noise to the target side of the dev set to generate synthetic training data (Meng et al., 2020); 2) Use multiple models to generate synthetic data through beam search decoding, and then add synthetic data to the dev test for fine-tuning.

Model ensemble is also a widely used technique in previous WMT workshops (Garmash and Monz, 2016), which can boost the performance by combining the predictions of several models at each decoding step. We selected the best four models from the six we trained for ensemble.

3 Settings

3.1 Experiment Settings

We use the open-source fairseq (Ott et al., 2019) for training, and use sacreBLEU (Post, 2018) to measure system performances instead of the BLEU script mentioned in the task. The main parameters are as follows: Each model is trained using 8 GPUs. The size of each batch is set as 2048, parameter update frequency as 32, learning rate as 5e-4 (Vaswani et al., 2017) and label smoothing as 0.1 (Szegedy et al., 2016). The number of warmup steps is 4000, and the dropout is 0.1. We employ joint sentencepiece model (Kudo and Richardson, 2018; Kudo, 2018) for word segmentation, with the size of the vocabulary set to 32k. Jieba tokenizer is used for Chinese word segmentation while Moses tokenizer for English and Russian word segmentation. The three languages share a vocabulary of 45K words. In the inference phase, we use the open-source marian (Junczys-Dowmunt et al., 2018) to perform decoding. The beam-size is 4 and the length penalty is set to 1.2.

System	BLEU
Data Filter	26.6
Multilingual model	29.3 (+2.7)
Full data denoise	30.0 (+0.7)
FTST + ru-zh denoise	31.9 (+1.9)
Ensemble	32.5 (+0.6)
2021 Final submit	27.7

Table 2: The experimental result of system

Training Strategy	Train Data	BLEU
Baseline	ru2zh	26.6
Enhanced target	+en2zh	28.7 (+2.1)
Enhanced target and source	+ru2en	29.3 (+0.6)
All Direction	+zh2ru zh2en en2ru	29.2 (-0.1)

Table 3: The experimental result of Multilingual Model

3.2 Training Process

We combine multi-stage denoising training with data augmentation methods. Figure 1 illustrates our training process:

1) We cleanse the training data using methods mentioned in 2.2.1 and train three forward models and one backward model.

2) We further denoise data using LaBSE (as mentioned in 2.2.3) and conduct denoising training until the model converge on the dev set.

3) We perform data augmentation as described in 2.2.2. We collect a total of 45M Russian monolingual data and split them into three sets, each with 15M sentences. We use three different forward models to generate three sets of training data. Hoping to add diversity to incremental training, we use the data synthesized by one model to train the other two models. For example, we use the synthetic data generated by forward model A to incremental train forward model B, C and so on. We also collect a total of 15M Chinese monolingual data and back translate the data using the backward model. We repeat back translation for three times and obtain three sets of back translation data. We incrementally train six models using the above synthetic data.

4) We average the last 5 checkpoints of each model and select the best four from the six models we trained for final ensemble.

4 Experiment Result

Our overall training strategy is to train a baseline model, conduct incremental training with techniques such as multilingual model, denoise training, data augmentation, and fine-tuning. Our submitted results come from ensembled models. Table 2 lists the results of our submission on dev set. Comparing with the baseline model, our final submission achieves an increase of 5.9 BLEU. Our baseline model is trained with data processed with methods mentioned in section 2.2.1. The BLEU score of the baseline model on the dev set is 26.6. Comparing with the baseline model, our multilingual strategy leads to a huge improvement of 2.7 BLEU. Our simplified denoising training strategy contributes to an increase of 0.7 BLEU. It should be noted that data augmentation techniques (FTST method and LaBSE denoising on ru2zh data) also result in a significant increase of 1.9 BLEU. Finally, an increase of 0.6 BLEU is gained via ensemble. Our submitted system gain 32.5 BLEU on the dev set, which demonstrate the effectiveness of our multiple strategies. According to the organizer's feedback, our submitted model gains 27.7 BLEU on the WMT21 test set.

5 Analysis

5.1 Multilingual Model and Model Performance

Our experiment results demonstrate that multilingual model has positive effects on system performance. We have experimented on different multilingual models and compare their results. Table 3 lists the results of different multilingual models. Compared with the baseline model, the multilingual model obtains 2.1 BLEU increase after adding en2zh data for training. A further 0.6 BLEU is achieved after adding the ru2en data, demonstrating that adding Russian data at the source side can optimize the encoder.

However, our experiment shows no improvement after adding data of other three directions. We adopt the enhanced target and source strategy for faster training, as training with all data might be considerably slow.

Training Strategy	BLEU
Baseline	26.6
+ru2zh denoise	28.0 (+1.4)
Enhanced target and source	29.3
+Full-data denoise	30.0 (+0.7)
+ru-zh denoise	30.5 (+0.5)

Table 4: The experimental result of denoising training

Training Strategy	BLEU
Enhanced target and source	29.3
Sampling BT	30.0
Beam BT	29.7
FT	29.7
Pivot FT	29.5
FTST	30.5

Table 5: The experimental result of data augmentation

5.2 Denoising Training and System Performance

Our experiment also demonstrates the contribution of denoising training to system performance. Table 4 compares the results of baseline and denoising training model, from which we can see an increase of 1.4 BLEU. We further compare the results measured at the three stages of denoising training. We use the enhanced target and source model to conduct simplified denoising training. Our experiment shows that full-data denoising training leads to an increase of 0.7 BLEU while ru2zh data denoising further leads to an increase of 0.5 BLEU. The experimental results show that the denoise strategy is effective and can lead to at least 1 BLEU improvement even after multilingual model enhancement.

5.3 Data Augmentation and System Performance

Data augmentation strategy also leads to huge BLEU improvements. We try multiple data augmentation strategies, including back translation (BT), forward translation (FT), FTST (2.2.2). Sampling BT means sampling from the model conditional distribution and beam BT means using beam search, when generating synthetic data. Table 5 shows the effects of different data enhancement methods. Our results show that sampling back translation can lead to better results (about 0.3 BLEU in our experiment). We also conduct two forward translation experiments: FT is translating Russian to Chinese directly, and Pivot FT is using English as the pivot language, which achieve only an undesirable result. We then using the FTST method and gain the best result with a BLEU score of 30.5. The experimental results show that the combination of sampling BT and FT data (FTST) can produce the best data augmentation effect.

6 Conclusion

This paper presents HW-TSC's submission to WMT21 Triangular Machine Translation Task. In general, we use Transformer architecture and explore multiple data filtering and selection methods. In terms of training and data processing strategies, multilingual model, denoising training, data augmentation, and FTST we used can effectively improve system performance. Our final result achieves an increase of 5.9 BLEU when comparing baseline model on the dev set and gain a BLEU score of 27.7 on the test which is the highest among all submissions.

References

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM model 2. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA*, pages 644–648.

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, Brussels, Belgium, October 31 - November 4, 2018*, pages 489–500.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2020. Language-agnostic bert sentence embedding. *arXiv preprint arXiv:2007.01852*.

Ekaterina Garmash and Christof Monz. 2016. Ensemble learning for multi-source neural machine translation. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1409–1418.

Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov.

2016a. Fasttext.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016b. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, et al. 2018. Marian: Fast neural machine translation in c++. *arXiv preprint arXiv:1804.00344*.

Huda Khayrallah and Philipp Koehn. 2018. On the impact of various types of noise on neural machine translation. *arXiv preprint arXiv:1805.12282*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 66–75.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018: System Demonstrations, Brussels, Belgium, October 31 - November 4, 2018*, pages 66–71.

Fandong Meng, Jianhao Yan, Yijin Liu, Yuan Gao, Xianfeng Zeng, Qinsong Zeng, Peng Li, Ming Chen, Jie Zhou, Sifan Liu, et al. 2020. Wechat neural machine translation systems for wmt20. *arXiv preprint arXiv:2010.00247*.

Myle Ott, Michael Auli, David Grangier, and Marc'Aurelio Ranzato. 2018. Analyzing uncertainty in neural machine translation. In *International Conference on Machine Learning*, pages 3956–3965. PMLR.

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. *arXiv preprint arXiv:1904.01038*.

Matt Post. 2018. A call for clarity in reporting bleu scores. *arXiv preprint arXiv:1804.08771*.

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, WMT 2019, Florence, Italy, August 1-2, 2019 - Volume 2: Shared Task Papers, Day 1*, pages 374–381.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826.

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 *Advances in neural information processing systems*, pages 5998–6008.

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. *arXiv preprint arXiv:1906.01787*.

Wei Wang, Taro Watanabe, Macduff Hughes, Tetsuji Nakagawa, and Ciprian Chelba. 2018. Denoising neural machine translation training with trusted data and online data selection. *arXiv preprint arXiv:1809.00068*.

Lijun Wu, Yiren Wang, Yingce Xia, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. 2019. Exploiting monolingual data at scale 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 4207–4216.