Papago's Submissions to the WMT21 triangular Translation Task

Jeonghyeok Park*
Papago, Naver Corp.
1170339900l1@sjtu.edu.cn

Hyunjoong Kim
Papago, Naver Corp.
soy.lovit@navercorp.com

Hyunchang Cho
Papago, Naver Corp.
hyunchang.cho@navercorp.com

Abstract

This paper describes Naver Papago's submission to the WMT21 shared triangular MT task to enhance the non-English MT system with tri-language parallel data. The provided parallel data are Russian-Chinese (direct), Russian-English (indirect), and English-Chinese (indirect) data. This task aims to improve the quality of the Russian-to-Chinese MT system by exploiting the direct and indirect parallel resources. The direct parallel data is noisy data crawled from the web. To alleviate the issue, we conduct extensive experiments to find effective data filtering methods. With the empirical knowledge that the performance of bilingual MT is better than multi-lingual MT and related experiment results, we approach this task as bilingual MT, where the two indirect data are transformed to direct data. In addition, we use the Transformer, a robust translation model, as our baseline and integrate several techniques, averaging checkpoints, model ensemble, and re-ranking. Our final system provides a 12.7 BLEU points improvement over a baseline system on the WMT21 triangular MT development set. In the official evaluation of the test set, ours is ranked 2nd in terms of BLEU scores.

1 Introduction

We participate in the WMT21 triangular machine translation task, using the direct and indirect parallel data to improve Russian-to-Chinese machine translation. The provided data consists of one noisy web corpus (Russian-Chinese, direct translation) and two combined bitexts from several public resources (English-Chinese/Russian, indirect). Such cases frequently occur in both actual translation services and research. In particular, this task is crucial in scenarios where we need to improve the performance of non-English translations or low-resource languages with high-resource parallel data.

Previous works deal with the triangular MT using several methods such as pivot-translation (Cheng et al., 2017), transfer learning (Kim et al., 2019), pre-trained multi-lingual MT (Liu et al., 2020; Tang et al., 2020) and so on.

In this paper, we explore existing novel techniques to integrate them for the triangular MT tasks. The original direct parallel degrades the translation quality of the model due to the noisy parts containing not well-aligned sentence pairs, erroneous characters, or the wrong language ID. To discard the noise parts of the noisy web corpus, we filter out data with sequence length, length ratio, language ID, de-duplication, and the sentence similarity computed with pre-trained multi-lingual language model (LaBSE, Feng et al., 2020). In preliminary experiments, we approached this task in three main ways: bilingual MT, multi-lingual MT, and fine-tuning the pre-trained multi-lingual translation model (i.e., mBART). As shown in Section 4, we found that the bilingual MT outperforms the others. Thus, to augment the Russian-to-Chinese corpus, we conduct two types of data augmentation: (1) back-translation on the discarded monolingual Chinese data from noise-refining steps and (2) translation using English as pivot language on two indirect bilingual data (e.g., feed English of English-Chinese data to English-to-Russian translation model to augment Russian-Chinese data). In detail, we generate the synthetic data by using different decoding methods such as beam search, sampling, and adding noise to beam search outputs. Our submission systems use 12-layer Transformer architecture. Furthermore, we exploit ensemble, averaging checkpoints, and noisy-channel re-ranking techniques to mitigate the over-fitting problem or improve the generalization capability in the test set.

To find suitable methods for triangular MT, we conduct extensive experiments, where all neural machine translation (NMT) systems are evaluated against the development set released in the WMT21

* Work done during internship at Naver Corp.

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 341–346

triangular MT shared task. Our final submission improves about 12.7 and 8.9 BLEU points compared to the organizer's baseline system on the development set and test set, respectively.

2 Approaches

2.1 Data Pre-processing

On all three corpora, we apply data normalization such as unifying punctuation marks and parentheses. For Chinese, we convert the traditional Chinese to Simplified Chinese using the open-source toolkit HanziConv[1] For all languages, we apply a language-specific tokenizer as a pre-tokenization step. We use NLTK[2] for English and Russian, jieba[3] for Chinese. And then, we apply joint multilingual Byte-Pair Encoding (BPE, Sennrich et al. 2016) to the pre-tokenized corpus with 75K merge-operations and 10K character limitation using the open-source toolkit Transformers[4].

2.2 Data Filtering

The provided parallel corpus contains a certain amount of noisy parts, which affects the translation quality. Thus, we eliminate noisy parts with the following heuristics rules:

- *Filtering out sentence pairs containing more than 256 tokens.
- *Filtering out sentence pairs consisting of characters of other languages than a predefined threshold. For this sake, we use an in-house language detector. We determine the threshold experimentally.
- *Filtering out sentence pairs with source/target length ratio exceeding 1.5 (Ott et al., 2019).
- Filtering out duplication in corpora (Khayrallah and Koehn, 2018; Ott et al., 2019). There are 4 options as follows: filtering out (1) duplicate sentence pairs (It is called Pair-dedup. in Table 3); (2) duplicate source sentences (Src-only-dedup.); (3) duplicate target sentences (Tgt-only-dedup.); (4) duplicate source and duplicate target sentences (Src&Tgt-dedup.).
- Filtering out sentence pairs according to the cosine similarity of the sentence pair. To this end, we feed the sentence pair to LaBSE (Feng et al., 2020) and calculate the cosine similarity score of the sentence pair. Then, we discard sentence pairs whose cosine similarity score falls below a certain threshold. From here on, it is called LaBSE filtering,

where the filtering methods marked with * are basic filtering methods.

We conducted experiments on the de-duplication and LaBSE filtering to find an optimal combination of them in subsection 4.1. Based on the results, we remove the duplicate sentence pairs and set the threshold of LaBSE filtering to 0.5 in our experiment. Table 1 shows the amount of the sentence pairs after filtering.

Systems	RU-ZH	EN-ZH	RU-EN
Original	33M	28M	69M
+ Basic Filter	22M	19M	50M
+ De-duplicate	18M	15M	42M
+ LaBSE Filter	13M	12.7M	39.3M

Table 1: The amount of the sentence pairs

2.3 Data Augmentation

To augment the direct bilingual data (Russian-to-Chinese), we generate synthetic bilingual sentence pairs on three data: one monolingual data (Chinese), two indirect parallel data (English-Chinese, and Russian-English). The Chinese monolingual corpora filtered out in the filtering step are translated back to Russian by the Chinese-to-Russian translation model (back-translation). To utilize indirect parallel data, we first train English-to-Chinese and English-to-Russian translation systems using provided corpora. Then we acquire synthetic Russian-Chinese pairs translating English sentences of English-Chinese data to Russian sentences using the trained English-to-Russian MT system (back-translated synthetic corpus). In the same way as before, we also acquire synthetic Russian-Chinese pairs translating English sentences of Russian-English data to Chinese using the English-to-Chinese MT system (forward-translated synthetic corpus). In this paper, we use the Transformer-Big model to augment the direct bilingual. In the future, we would thoroughly ex-

[1]https://github.com/berniey/hanziconv
[2]https://www.nltk.org
[3]https://github.com/fxsjy/jieba
[4]https://github.com/huggingface/transformers

plore several methods to improve further the quality of augmented data, such as using a bigger model and iterative back-translation Hoang et al. (2018).

Following Edunov et al. (2018), we use various decoding strategies , including maximum a-posteriori (MAP) and non-MAP methods for effective data augmentation. In detail, there are five decoding methods: (1) beam search; (2) sampling; (3) sampling top 10; (4) noising beam outputs; (5) noising beam outputs only with sentences longer than 5. The sampling top 10 method is to restrict the sampling method to the k highest-scoring outputs at every decoding step. The noising beam outputs method denotes to add three types of noise such as random permutation over tokens, deleting some tokens, and masking some tokens. Edunov et al. (2018) demonstrated that the non-MAP decoding methods such as (2)-(5) outperform pure beam search.

In our experiments, we generate the five synthetic bilingual data using the different decoding methods. We train five models with each synthetic data embracing data variation. Performance of each way is described in Table 4 In the final submission, we choose a combination of (1) beam search, (2) sampling, and (3) restricted noising beam outputs experimentally. After generating the synthetic bilingual data, we apply the data filtering schemes described in section 2.2 to them. We upsample bitext data to maintain a 1-to-1 ratio of real to synthetic bitext during the training phase.

2.4 Model

In our experiments, we adopt three Transformer architectures.

- **Transformer-Base** with a 6-layers encoder-decoder and a model dimension of 512 as used in Vaswani et al. (2017).
- **Transformer-Big** with a 6-layers encoder-decoder and a model dimension of 1024 as used in Vaswani et al. (2017).
- **Transformer-Large** is similar to Transformer-Big model except that it uses a 12-layers encoder-decoder with pre-norm (Wang et al., 2019).

To boost the performance of the translation model, we average the parameters acquired from various epochs obtained in a training phase and then ensemble the averaged checkpoints involving various variations in terms of data. Moreover, we perform a grid search for decoding hyper-parameters such as length penalty and beam size to find the best performance. We conduct preliminary experiments using the Transformer-Big model to find (sub)optimal configurations in data filtering, data augmentation, hyper-parameters, and so on. Then, based on the observations, we apply the (sub)optimal configurations to the Transformer-Large model.

2.5 Noisy-Channel Re-ranking

The noisy channel re-ranking (Yee et al., 2019) applies Bayes' rule to decoding:

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)}, \tag{1}$$

where x is source sequence and y is hypothesis sequence in translation task. Since $p(x)$ is constant for all y, re-ranking score for each hypothesis candidate can be reconstruct as follows:

$$\frac{\lambda_1 \log p(y|x) + \lambda_2 \log p(x|y) + \lambda_3 \log p(y)}{|y|^\alpha}, \tag{2}$$

where λ, α are tunable weights, $|y|$ is length of hypothesis sequence, and $p(y|x)$, $p(x|y)$, $p(y)$ denote score of forward model, backward model, and language model, respectively.

In a preliminary experiment, we used several publicly released Chinese language models [5] and found that they caused performance degradation. In addition, we used another scoring metric (inverse document frequency similar to BARTScore (Yuan et al., 2021)) when re-ranking, but this did not give any performance gain. Due to time and resource constraints, we could not fully explore our own Chinese language model.

3 Experiments and Results

3.1 Experiment Setup

Our base system is based on the Transformer-Large with an embedding size of 1024, 12 encoder and decoder layers, 12 attention heads, shared source and target embedding, the sinusoidal positional embedding, and pre-norm. We train with a batch size of 3584 tokens and optimize the model parameters using Adam optimizer with a learning rate 1e-3 β_1= 0.9 and β_2 = 0.98, learning rate warm-up

[5] https://huggingface.co

Systems	#Sentence	BLEU
Organizer's Systems		
Direct	33M	20.2
Pivot	(69+28)M	19.7
Our Systems		
Transformer-Base	22M*	26.4
Transformer-Big	22M*	28.7
Transformer-Large	22M*	28.9
+ De-duplication	18M	29.2
+ LaBSE Filter (0.5)	13M	29.8
+ Augmented data	(13+45)M	31.3
+ Averaging	-	31.9
+ Ensemble	-	33.0
+ Re-ranking**	-	33.0

Table 2: Performances on the WMT21 triangular MT Russian-to-Chinese development set in Transformer-Large. The asterisk(*) mark is a basic filter, and the double-asterisk(**) denotes our submitted system.

Systems	#Sentence	BLEU
Basic filter	22M	28.7
LaBSE filter (0.5)	17M	29.5 (+0.8)
Pair-dedup.	18M	28.7 (+0.0)
+ LaBSE filter	13M	29.7 (+0.9)
Src-only-dedup.	12.6M	28.5 (-0.3)
Tgt-only-dedup.	13M	28.5 (-0.2)
Src&Tgt-dedup.	11M	28.9 (+0.1)
+ LaBSE filter	10.6M	29.8 (+1.0)

Table 3: Further experiment results with different data filtering methods in Transformer-Big. The Basic filter contains filtering the sentences by sequence length, language ratio, and length ratio. The others are described in section 2.2. The plus marks denote that the filtering method is applied additionally. For example, the "+ LaBSE filter" in fifth row means that both Pair-dedup and LaBSE filter are applied.

over the first 16k steps. Additionally, we apply label smoothing with a factor of 0.1. In the training phase, the dropout is set to 0.1, and the attention dropout is set to 0.3. We apply the early stopping technique using the WMT21 triangular MT development set, and all models are trained for a minimum of 30 and a maximum of 50 epochs. We trained all our models using FAIRSEQ [6] (Ott et al., 2019) on 8 NVIDIA Tesla V100 GPUs.

3.2 Experimental Results

As shown in Table 2, our final model outperforms about +12.7 BLEU compared to the organizer's systems. In detail, we got the most significant performance improvement in scaling up the Transformer model. Through the data filtering process, our model achieved an improvement of about 1 BLEU. By augmenting the Russian-to-Chinese corpus, our model obtained a gain of about 1.5 BLEU. When model-level methods such as averaging parameters, ensemble, and re-ranking were applied, the BLEU score could be raised again by 1.5.

4 Discussions

4.1 Analysis of Data Filtering

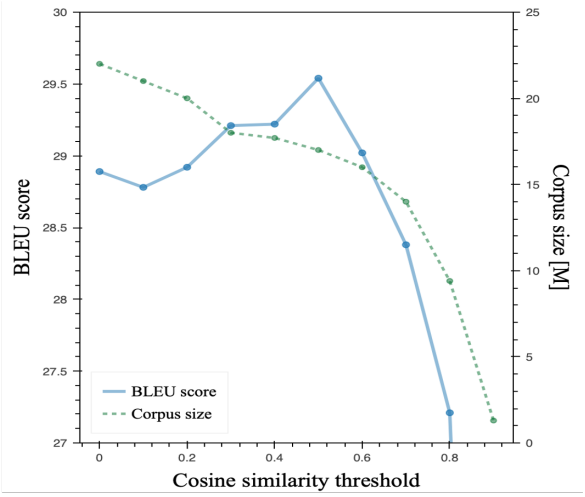
In order to verify the impact of various data filtering methods on translation performance, we conduct experiments on the direct parallel corpus (i.e., Russian-to-Chinese). As shown in Table 3, we can see that the performance improves even though we have removed more than half of the data, which means that the original data is quite noisy.

To find the best threshold value for the LaSBE filtering, we executed an additional experiment in which the threshold range is set from 0.0 to 0.9. The threshold 0.0 denotes that the filtering is not applied, and threshold 0.9 means filtering out the sentence pairs whose cosine similarity score falls below 0.9. As can be seen from the results in Figure 1a, we set the threshold value of LaBSE filtering to 0.5 in our final system. Figure 1b shows the distribution of cosine similarity scores on training data. In contrast to the distribution of the train data, that of the WMT21 triangular MT development set is clustered around 0.8. It means that the train data contain many noisy sentence pairs (nearby cosine similarity score 0.2) in terms of LaBSE sentence similarity.

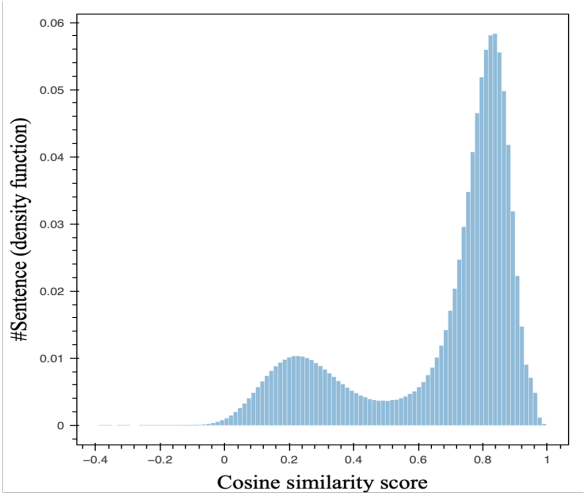
4.2 Analysis of Data Augmentation

We evaluate the impact of the different decoding methods for data augmentation. Table 4 shows the experiment results, which are consistent with Edunov et al. (2018). We observed that sampling and noise beam search are more effective than vanilla beam search. In particular, it is more effective to limit adding noise only to sentences longer than 5 (Noising beam*). As shown in the Table 4, none of the decoding strategies demonstrates superior performance. Therefore we ensemble models

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



(a) Performances with different threshold values



(b) Cosine similarity scores on train data

Figure 1: The LaBSE filtering methods.

Systems	#Sentence	BLEU
Before augment.	13M	29.6
Beam	(13+45)M	30.2 (+0.5)
Sampling	(13+54)M	30.7 (+1.0)
Sampling top 10	(13+47)M	30.5 (+0.9)
Noising beam	(13+45)M	30.3 (+0.6)
Noising beam*	(13+45)M	30.8 (+1.1)

Table 4: Further experiment results with different decoding methods for data augmentation in Transformer-Big. The Before augment. denotes applying the basic filtering, de-duplication (pair), and LaSBE filtering to data. The asterisk(*) mark denotes the restriction to sentences with the length of tokens longer than 5.

Systems	Data	BLEU
Transformer-Large	RU2ZH*	31.3
mBART50	RU2ZH	30.3 (-1.0)
mBART50	RU2ZH*	30.9 (-0.4)
mBART50	M2ZH	29.4 (-1.9)
mBART50	M2M	30.3 (-1.1)

Table 5: Comparison between Transformer trained from scratch and fine-tuned mBART50 in an aspect of BLEU score. The asterisk(*) mark denotes augmentation with noising beam search. The M2ZH and M2M indicate RU2ZH&EN2ZH and RU2ZH&EN2ZH&RU2ZH, respectively.

trained with the different decoding methods. As a result, the ensemble model performs better, as seen in the Table 2. In an additional experiment, we also find that the performance of the augmentation (back-translation) models has a significant impact on the performance of the forward model as suggested in Hoang et al. (2018)

4.3 Bilingual MT vs Multi-lingual MT

We experimented with two ways to fully utilize the triangular MT data: to transform the indirect parallel data into direct parallel data and use them for bilingual MT as described in subsection 2.3; the another is to use the all provided data for multi-lingual MT. From the experiment result, we observed that the bilingual MT outperforms the multi-lingual MT by 1 BLEU point, and the multi-lingual MT requires more training time due to upsampling specific direction data.

4.4 Pre-trained Multi-lingual Language Model

Recently, fine-tuning pre-trained multi-lingual MT models (Liu et al., 2020; Tang et al., 2020) showed remarkable performance in multi-lingual translation scenarios. To explore the effectiveness of fine-tuning a pre-trained multi-lingual translation model for triangular MT, in the Table 5, we conducted experiments using mBART50 (Tang et al., 2020) on several datasets: (1) the augmented RU2ZH with beam search; (2) the augmented RU2ZH with noising beam search; (3) the RU-ZH and EN-ZH data; (4) the RU-ZH, EN-ZH, and RU-EN data. We use the Transformer-Large equal to mBART50 in model size as a baseline model for a fair compar-

ison. For fine-tuning mBART, the WMT21 triangular MT development set is used to compute the stopping criterion, and the models are fine-tuned for a minimum of 10 and a maximum of 20 epochs. In general, mBART transfer learning is known to be effective in low-resource language data. Fine-tuning mBART does not work well when large enough data are available. As can be seen from the experimental results, it is more effective to train the model from scratch after data augmentation.

5 Conclusion

This paper depicts Papago's submissions to the WMT21 triangular MT shared task. We have conducted extensive experiments using various techniques such as data filtering, data augmentation, model ensembling, and re-ranking in the triangular MT scenario. Except for existing techniques, we also have tried to apply data filtering with LaBSE sentence score and data augmentation using pivot language and demonstrated their effectiveness in translation performance. As a result, our system achieves the second record according to the released official results.

References

Yong Cheng, Qian Yang, Yang Liu, Maosong Sun, and Wei Xu. 2017. Joint training for pivot-based neural machine translation. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 3974–3980.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. *CoRR*, abs/1808.09381.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2020. Language-agnostic BERT sentence embedding. *CoRR*, abs/2007.01852.

Vu Hoang, Philipp Koehn, Gholamreza Haffari, and Trevor Cohn. 2018. Iterative back-translation for neural machine translation. pages 18–24.

Huda Khayrallah and Philipp Koehn. 2018. On the impact of various types of noise on neural machine translation. *CoRR*, abs/1805.12282.

Yunsu Kim, Petre Petrov, Pavel Petrushkov, Shahram Khadivi, and Hermann Ney. 2019. Pivot-based transfer learning for neural machine translation between non-english languages. *CoRR*, abs/1909.09524.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *CoRR*, abs/2001.08210.

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. *CoRR*, abs/1904.01038.

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.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2020. Multilingual translation with extensible multilingual pretraining and finetuning. *CoRR*, abs/2008.00401.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

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. *CoRR*, abs/1906.01787.

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.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. Bartscore: Evaluating generated text as text generation. *CoRR*, abs/2106.11520.