HW-TSC's Participation in the WMT 2020 News Translation Shared Task

Daimeng Wei[1], Hengchao Shang[1], Zhanglin Wu[1], Zhengzhe Yu[1], Liangyou Li[2], Jiaxin Guo[1], Minghan Wang[1], Hao Yang[1], Lizhi Lei[1], Ying Qin[1], Shiliang Sun[3],
[1]Huawei Translation Service Center, Beijing, China
[2]Huawei Noah's Ark Lab, Hong Kong, China
[3]East China Normal University, Shanghai, China
{weidaimeng,shanghengchao,wuzhanglin2,yuzhengzhe,liliangyou,
guojiaxin1,wangminghan,yanghao30,leilizhi,qinying}@huawei.com
slsun@cs.ecnu.edu.cn

Abstract

This paper presents our work in the WMT 2020 News Translation Shared Task. We participate in 3 language pairs including Zh/En, Km/En, and Ps/En and in both directions under the constrained condition. We use the standard Transformer-Big model as the baseline and obtain the best performance via two variants with larger parameter sizes. We perform detailed pre-processing and filtering on the provided large-scale bilingual and monolingual dataset. Several commonly used strategies are used to train our models such as Back Translation, Ensemble Knowledge Distillation, etc. We also conduct experiment with similar language augmentation, which lead to positive results, although not used in our submission. Our submission obtains competitive results in the final evaluation.

1 Introduction

This paper introduces our work for the WMT 2020 News Translation Shared Task. We participate in three language pairs including Chinese/English (Zh/En), Khmer/English (Km/En), Pashto/English (Ps/En) and in both directions. After observation, we consider that the officially provided dataset has the acceptable size and quality therefore only participate in the constrained evaluation. Our method is mainly based on previous works but with fine-grained data cleaning techniques and language pair specific optimizations.

For each language pair, we perform careful multi-step cleaning on the provided dataset and only keep a high-quality subset for training. At the same time, several strategies are tested in a pipeline including Back-Translation (Edunov et al., 2018), Ensemble Knowledge Distillation (Freitag et al., 2017; Li et al., 2019), Forward Translation (Wu et al., 2019), Fine-Tuning (Sun et al., 2019), and Ensemble and Re-ranking (Ng et al., 2019a).

Due to the page limitation, we mainly introduce our methods and experiments on the Zh-En and En-Zh language pairs. Most of these methods are also employed on the Km/En and Ps/En pairs. Special optimizations regarding different language will be introduced separately.

2 Data

In this section, we describe the size and source of the dataset as well as our cleaning and filtering techniques.

2.1 Data Source

2.1.1 Zh/En

We use both bilingual and monolingual text to train the model. Regarding bilingual text, we merge the data from CCMT (7M), Wiki Titles v1 (1M), News Commentary v15 (0.4M) and a subset of UN Parallel Corpus (9M). We also select 10 million of Zh and En monolingual text from Xin Hua, XMU and News crawl respectively for back translation.

2.1.2 Km/En

We use the Para Crawl v5.1 (4.17M), Khmer and Pashto parallel data (0.29M) as the bitext corpus, and select 10M monolingual text from Common Crawl and news crawl 2018 for Km and En, respectively.

2.1.3 Ps/En

Similar to Km/En, we also use the Para Crawl v5.1 (1M), Khmer and Pashto parallel data (0.03M) as bitext and select 6.5M monolingual text from Common Crawl and news crawl 2018.

2.2 Data Pre-processing

For the Zh/En corpus, we use following operations to pre-process the data:

Proceedings of the 5th Conference on Machine Translation (WMT), pages 293–299
Online, November 19–20, 2020.

Operation	Zh-En			Km-En			Ps-En		
	Zh-En (bi)	Zh (mono)	En (mono)	Km-En (bi)	Km (mono)	En (mono)	Ps-En (bi)	Ps (mono)	En (mono)
Original	21	21.4	18	4.46	12.59	10	1.05	6.6	4.71
+ Deduplication	20.9	21.3	17.9	4.30	12.57	9.99	1.05	5.99	4.70
+ Lang-id filtering	20.4	19.6	17.9	2.82	11.13	9.93	1.02	5.69	4.38
+ Length filtering	20.1	19	17.9	2.71	10.54	9.90	0.94	4.97	4.14
+ Fast-align filtering	19.5	-	-	0.8	-	-	0.54	-	-
+ Data-selection	16.5	10	10	-	-	-	-	-	-

Table 1: This table shows the remaining data size of performing specific data cleaning and selection operations, where the unit is million (M). The bilingual (bi) and monolingual (mono) texts are both listed in the table for all three language pairs.

- Regarding Chinese text, we tokenize the text with Jieba[1] tokenizer, and create the BPE (Sennrich et al., 2016) vocab with 30K merge operations.
- For English text, we use moses[2] tokenizer and generate a BPE vocab with 32K merge operations.
- Bitexts with length ratios (source/target) greater than 3 are removed.
- Texts longer than 120 sub-tokens are removed.
- Texts with undesired fastText-langid (Joulin et al., 2016b,a) are removed.

For the Km/En and Ps/En corpus, following operations are performed on the data:

- Full-width texts are converted to half-width texts.
- De-duplication is performed.
- Texts which the source or target is empty are empty.
- Sentences with undesired fastText-langid (Joulin et al., 2016b,a) are removed.
- SPM with regularization (Kudo and Richardson, 2018; Kudo, 2018) is used for both language pairs.
- Fast-align (Dyer et al., 2013) is used to further clean the corpus.
- Sentences with more than 100 sub-tokens are removed.

[1]https://github.com/fxsjy/jieba
[2]http://www.statmt.org/moses/

During experiment, we notice that Km and Ps data has relatively low qualities, which need to be further cleaned in a stricter manner. Therefore, we gradually increase the threshold of fast-align, and remove about 50% of un-aligned text to improve the training data quality. Detailed data size of each step is shown in Table 1.

2.3 Data Selection

Data selection filters out bilingual or monolingual out-of-domain text from a given corpora. We perform data selection on the Zh/En UN dataset, of which the domain is different from news. To do so, we train a classifier to select texts classified as news from the UN corpus. In terms of the classifier, when selecting En→Zh bi-text, we sample the target language (Zh) text from UN and non-UN dataset with an equal size (e.g. 50000), and label them with UN and news tags. Then, we train a Fasttext (Bojanowski et al., 2017) classifier on the sampled set, and score the leftover UN set with the classification probability $P(y = \text{news}|x)$ to retrieve the top-k bi-text pairs, where k is set to 9M in the experiment. Note that even if the score is lower than 0.5, we still keep the sample if its rank is within top-k. This method is also used for Zh→En selection. Note that the selected En→Zh and Zh→En set can be overlapped but not exactly the same.

From the experiment, we find that data selection is quite effective in improving the BLEU score on WMT 2019 test set compared to using entire UN set with a 1.1 increase on Zh→En and a 1.6 increase on En→Zh, respectively.

For the Km/En and Ps/En pairs, we do not employ the data selection strategy, but carefully evaluate the performance of different sources in the training set and finally select the Common Crawl (Km) and News Crawl (En) as the monolingual corpus. KenLM (Heafield, 2011) is also used to filter the data.

3 System Overview

This section describes the model and techniques of our work. We basically perform such strategies sequentially. Our experimental result will be presented on each part.

3.1 Model

Transformer (Vaswani et al., 2017) has been widely used for machine translation in recent years, which has achieved good performance even with the most primitive architecture without much modifications. Therefore, we choose to start from Transformer-Big and consider it as a baseline. Two variants of Transformer are also evaluated during the experiments, which are the model with wider FFN layers proposed in (Ng et al., 2019b), and the deeper encoder version proposed in (Sun et al., 2019). Here, we call two variants Transformer-Large and Transformer-Deep. Our models are implemented with THUMT (Zhang et al., 2017), and trained on a platform with 8 V100 GPUs.

3.2 Back Translation

Following (Edunov et al., 2018), we use back translation (BT) to improve the system performance. However, unlike (Edunov et al., 2018), we use beam search to decode the pseudo source text because in the experiment we find that results from beam search is better than sampling.

To acquire better monolingual text, we also use the method introduced in the data selection section to filter the in-domain subset for BT. For Zh→En and En→Zh direction, we use texts in target language from our bilingual corpus as the in-domain set, monolingual corpus as the out-of-domain set to train the classifier, and finally select approximately 10 million of samples for each direction. The back translated corpus are merged with the original corpus, which improves the performance by 0.6 for Zh→En and 1.3 for En→Zh. For the Km/En pair, we use exactly the same method as Zh/En, but with monolingual corpus from specific language, resulting in improvements of 5.33 and 2.55 in terms of BLEU for Km→En and En→Km on the devtest 20. For Ps/En, BT is performed on the selected data described in previous section, achieving improvements of 8.08 (Ps→En) and 2.89 (En→Ps) in terms of BLEU on each direction.

3.3 Ensemble Knowledge Distillation

Ensemble Knowledge Distillation (Freitag et al., 2017; Li et al., 2019) improves the performance of a student model by distilling knowledge from a group of trained teacher model into it. Comparing with some soft label distillation methods, the EKD for NMT is relatively straightforward, which can be implemented by training the student on the combination of the original training set and the translation from the ensembled teacher model on the training set. In our experiments we ensemble four models as the teacher model to translate the training set. Then, compute the BLEU for each sentence against the ground truth target. We keep 2/3 of the top scored translations for distillation and merge them into the original training set.

Generally speaking, EKD can be performed in an iteration manner. However, this could bring negative influence on the final ensemble. Therefore, we only do it once. EKD improves the BLEU by 1.5 points on the Zh→En direction, but only 0.2 points on the En→Zh direction.

We didn't perform the EKD on the Km/En and Ps/En pairs due to the limitation of the corpus size.

3.4 Forward Translation

As described in (Wu et al., 2019), similar to back translation, the monolingual corpus in source language can also be used to create the forward translation text with a trained MT model, and the created forward and backward translation corpus can both be merged with the original bilingual data. This strategy can enlarge the data size to a large extent. There are basically four steps to perform the forward translation. Take En→Zh as an example: 1) train M models with EKD in both direction; 2) create pseudo corpus with the ensemble of M models on the monolingual corpus in both direction (SRC→TGT', TGT→SRC'); 3) merge the created corpus with others (BT + FT + EKD + bilingual). 4) train a new model on the mixed corpus. This technique improves the performance by 1.0 in terms of BLEU on En→Zh direction and 0.4 BLEU on Zh→En direction. We also perform this strategy on Km/En and Ps/En, which achieves the improvements of 2.50 and 1.17 on En→Km and Km→En directions; 0.18 and 0.65 on En→Ps and Ps→En directions.

Note that the model trained with this technique can be ineffective for ensemble, which means such training strategy might decrease the model diver-

sity.

3.5 Fine-tuning

Previous works demonstrate that fine-tuning a model on in-domain data such as last year's test set could effectively improve the performance of this year (Sun et al., 2019). In the experiment, we fine-tune the model on the newstest18 for Zh→En with 3000 tokens per batch for one epoch, successfully achieving 3.6 of BLEU improvements on the newstest19. Furthermore, we keep the test corpus with orilang as Zh from newstest18 for fine-tuning, gaining an additional 1.0 BLEU increase. However, this method only obtains 0.2 BLEU increase on the En→Zh direction.

Km/En and Ps/En are newly introduced language pairs in the evaluation this year, thereby have no previous test sets. Since an additional devtest set is provided in addition to the dev set, we fine-tune models on the dev set and test on the devtest set. The experiment shows that fine-tuning could achieve 5.12 and 0.13 of improvements for En→Km and Km→En; 0.59 and 0.79 for En→Ps and Ps→En.

3.6 Ensemble

Six Transformer models are trained with different seeds, including 2 deep, 2 big and 2 large variants. The ensemble model improves the performance by 1.0 on Zh→En and 0.4 on En→Zh in terms of BLEU.

For Km/En and Ps/En pairs, we trained 4 and 6 Transformer-Deep models for Km/En and Ps/En. However, due to the size limitation, the improvements of ensemble is not significant for these two language pairs.

3.7 Ensemble MT Fine-tuning

We perform an additional experiment, named Ensemble MT Fine-tuning. First of all, we fine-tune 6 models on the 18 test set and produce the translation (mt) with the ensemble of them on the 19 test set. Then, we fine-tune the un-fine-tuned 6 models with the mt, which surprisingly improves about 0.6 BLEU on En→Zh. But we see no improvements on Zh→En. This experiment is also performed on Km/En and Ps/En language pairs, but only obtains limited improvements.

While submission, we fine-tune all 6 models on 18 test set and produce the mt with the ensemble model on the 20 test set. We then use the mt of 20 test set to fine-tune the original un-fine-tuned model to get the final one.

	Zh→En	
System	news2018	news2019
baseline	24.98	25.76
+ Data Selection	25.44	26.89 (+1.1)
+ Back-Translation	27.11	27.49 (+0.6)
+ EKD	27.18	29.06 (+1.5)
+ Forward-Translation	28.55	30.45 (+0.4)
+ Fine-tuning	-	35.07 (+4.6)
+ Ensemble	-	36.11 (+1.0)
+ Ensemble MT Fine-tune	-	36.11 (+0.0)
2020 Submission	34.3	

Table 2: The experimental result of Zh→En

	En→Zh	
System	news2018	news2019
baseline	37.84	34.86
+ Data Selection	38.91	36.47 (+1.6)
+ Back-Translation	44.29	38.48 (+1.3)
+ EKD	44.19	38.68 (+0.2)
+ Forward-Translation	43.79	39.69 (+1.0)
+ Fine-tuning	-	39.89 (+0.2)
+ Ensemble	-	40.32 (+0.4)
+ Ensemble MT Fine-tune	-	41.00 (+0.6)
2020 Submission baseline	46.0	

Table 3: The experimental result of En→Zh

3.8 Re-ranking

We also tested the noisy channel re-ranking proposed in (Ng et al., 2019b). However, we do not see consistent improvements on the news2019 and devtest set, thus we give up using the strategy in the submission for all three language pairs.

3.9 Similar Language Augmentation

We also investigate whether performing data augmentation with corpora in similar languages can boost system performances on low resource tasks like En/Km and En/Ps. Inspired by (Kudugunta et al., 2019) who propose the concept of language similarity that can be measured by the SVCCA score on hidden representations of a language pair.

We select top-two similar languages for Km and Ps, by referring to the (Kudugunta et al., 2019). We then collect a set of bilingual text from these languages and mix them into the original training set. For Ps, we collect bilingual corpus of Persian (Fa) and Urdu (Ur) for augmentation, and create

System	Km→En dev	devtest
baseline	7.54	5.90
+ Strict Fast-align	10.63	8.69 (+2.79)
+ Back-Translation	16.48	14.02 (+5.33)
+ Forward-Translation	18.04	15.19 (+1.17)
+ Fine-tuning	-	15.32 (+0.13)
+ Ensemble	-	15.47 (+0.15)
2020 Submission		25.33

Table 4: The experimental result of Km→En

System	En→Km dev	devtest
baseline	29.27	27.93
+ Strict Fast-align	41.39	37.72 (+9.79)
+ Back-Translation	44.61	40.27 (+2.55)
+ Forward-Translation	46.81	42.77 (+2.50)
+ Fine-tuning	-	47.89 (+5.12)
+ Ensemble	-	48.46 (+0.57)
2020 Submission		58.58

Table 5: The experimental result of En→Km. Note that the BLEU score of the dev and devtest are calculated with sentences tokenized with char-based tokenizer.

System	Ps→En dev	devtest
baseline	5.43	6.9
+ Strict Fast-align	7.4	7.31 (+0.41)
+ Back-Translation	14.96	15.39 (+8.08)
+ Forward-Translation	15.87	16.04 (+0.65)
+ Fine-tuning	-	16.83 (+0.79)
+ Ensemble	-	17.25 (+0.42)
2020 Submission		23.1

Table 6: The experimental result of Ps→En

a mixed corpora (Ps:Fa:Ur=8:10:3) with a size of 2.6M, much larger than the original bitext corpus. For Km, we collect Polish (Pl) and Corsican (Ca) as the augmentation language (Km:Pl:Ca=2:2:1) and mix them with the total size of 2.1M.

The experimental result shows that the augmentation improves the BLEU score by 1-3 points on all directions compared to merely training on the original training set, demonstrating that incorporate data of similar languages for data augmentation is effective. However, this advantage disappears when comparing with the strategy of using the Forward and Backward Translation with original language pair, because BT and FT fill the gap of the difference in the data size, and thereby fills the gap of the performance.

System	En→Ps dev	devtest
baseline	4.15	4.3
+ Strict Fast-align	6.0	6.13 (+1.83)
+ Back-Translation	9.01	9.02 (+2.89)
+ Forward-Translation	9.3	9.2 (+0.18)
+ Fine-tuning	-	11.02 (+0.59)
+ Ensemble	-	11.44 (+0.42)
2020 Submission		12.1

Table 7: The experimental result of En→Ps

Although this strategy works fine on a corpus with limited size, it is not as feasible as BT. At the same time, we understand that applying external similar language corpora is not allowed in the constrained track, and finally give up this method. But we would like to conduct further researches on this direction.

4 Results

This section presents the experimental results for each direction of all three language pairs in Table 2,3,4,5,6and 7, where the contribution of strategies introduced in previous sections are listed in each row.

5 Analysis

Here are several findings worthy of sharing during our experiments:

- We test different combinations of model architectures for ensemble, and find that the heterogeneous combinations often perform better than homogeneous combinations when the performance of each model is similar. We suppose that heterogeneous architectures are good at learning different kinds of patterns, which is potentially effective for ensemble.

- While performing data selection, we also test language models as described in (Ng et al., 2019b), but found that fasttext performed better than LMs. We consider this finding is

relatively intuitive because the objective of training the classifier could naturally distinguish features of inter-class samples and cluster inner-class samples, which should be more efficient than using LMs.

- When we perform back-translation and forward-translation on Km/En pairs, we find that no matter in which direction, monolingual text from news domain performs consistently better than that from wiki domain, but the bilingual texts are actually from wiki. The reason for the performance improvements contributed by news corpus might be that the size of the filtered bilingual corpus is small, therefore requires to learn more semantic patterns from BT and FT. Such semantic patterns appear more often in news corpus and thus surpass the loss caused by domain shifting.

6 Conclusion

This paper presents the submissions by HW-TSC on the WMT 2020 News Translation Task. For each direction in three language pairs, we perform experiments with a series of pre-processing and training strategies. The effectiveness of each strategy is demonstrated. Our experiments on similar language augmentation shows that corpora with similar languages can be used for performance improvements in low resource scenarios. Our submission finally achieves competitive result in the evaluation.

References

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Trans. Assoc. Comput. Linguistics*, 5:135–146.

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.

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

Kenneth Heafield. 2011. Kenlm: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation, WMT@EMNLP 2011, Edinburgh, Scotland, UK, July 30-31, 2011*, pages 187–197.

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*.

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.

Sneha Reddy Kudugunta, Ankur Bapna, Isaac Caswell, and Orhan Firat. 2019. Investigating multilingual NMT representations at scale. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 1565–1575.

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, WMT 2019, Florence, Italy, August 1-2, 2019 - Volume 2: Shared Task Papers, Day 1*, pages 257–266.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019a. Facebook fair's WMT19 news translation task submission. 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 314–319.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019b. Facebook FAIR's WMT19 News Translation Task Submission. *arXiv preprint arXiv:1907.06616*.

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, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*.

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.

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*, pages 5998–6008.

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 2019, Hong Kong, China, November 3-7, 2019*, pages 4205–4215.

Jiacheng Zhang, Yanzhuo Ding, Shiqi Shen, Yong Cheng, Maosong Sun, Huan-Bo Luan, and Yang Liu. 2017. THUMT: an open source toolkit for neural machine translation. *CoRR*, abs/1706.06415.