OPPO's Machine Translation Systems for WMT20

Tingxun Shi, Shiyu Zhao, Xiaopu Li, Xiaoxue Wang, Qian Zhang
Di Ai, Dawei Dang, Zhengshan Xue and **Jie Hao**
Manifold Lab, OPPO Research Institute, Beijing, China
{shitingxun, zhaoshiyu, lixiaopu, wangxiaoxue, zhangqian666,
aidi1, dangdawei, xuezhengshan, haojie}@oppo.com

Abstract

In this paper we demonstrate our (OPPO's) machine translation systems for the WMT20 Shared Task on News Translation for all the 22 language pairs. We will give an overview of the common aspects across all the systems firstly, including two parts: the data preprocessing part will show how the data are preprocessed and filtered, and the system part will show our models architecture and the techniques we followed. Detailed information, such as training hyperparameters and the results generated by each technique will be depicted in the corresponding subsections. Our final submissions ranked top in 6 directions (English ↔ Czech, English ↔ Russian, French → German and Tamil → English), third in 2 directions (English → German, English → Japanese), and fourth in 2 directions (English → Pashto and and English → Tamil).

1 Introduction

This paper describes the OPPO's submission to the Fifth Conference on Machine Translation (WMT20) news translation shared task. We built Transformer (Vaswani et al., 2017)-based systems for all the directions, and applied several well-known, widely-used techniques, such as large-scale back-translation (Sennrich et al., 2016a) and forward-translation, model ensemble and reranking. Since all the systems share a roughly similar data preprocessing and training methods, to avoid duplication words, we will demonstrate the common knowledge in Section 2 firstly, which will be divided into two parts: the preprocessing part shows the data preprocessing pipeline and data filtering pipeline, the latter is generally composed by rule-based filtering and alignment-based filtering; the training part depicts the techniques we applied. Detailed information, including training hyperparameters, the results generated by each technique, and some other explorations will be listed in each corresponding direction in Section 3. Finally, we will summarize the report and indicate our final works. We used *marian* (Junczys-Dowmunt et al., 2018) to implement our systems for English ↔ {Khmer, Russian, Tamil} and French ↔ German task pairs, and *fairseq* (Ott et al., 2019) for the rest [1].

2 System Overview

We preprocess corpora in two stages. In the preprocessing stage, data is converted but not filtered. The common pipeline of preprocessing including the following steps:

- Remove non-utf8 characters
- Unescape html characters, e.g. "&gt;" is converted to ">"
- Normalize different kinds of spaces and punctuations
- Tokenization
- True case

The last three steps are all processed by *moses* scripts. This pipeline is both applied for the parallel corpora and monolingual corpora, and true case models are generally trained on the mixture of parallel and monolingual datasets.

After preprocessing we filter the parallel corpora according to statistical information and alignment information, set the thresholds according to our previous experiences. For the statistic perspective, we mainly focus on some heuristic rules, contain but not limited in

[1] Choice on the training framework is only depends on personal habit.

Proceedings of the 5th Conference on Machine Translation (WMT), pages 282–292
Online, November 19–20, 2020.

- Pairs of which the source side and the target side are the same.

- Pairs contain blank lines.

- Pairs contain too long sentences (typically those have more than 200 words).

- Pairs that have abnormal source-target length ratios. The source-target length ratio is defined as the words count ratio between the source and the target. Typically the upper bound is 2.5 and the lower bound is 0.4.

- Pairs that have irregular character-word length ratios. The character-word length ratio is defined as the ratio between the count of characters and the count of words. Generally the upper bound is 12 and the lower bound is 1.5.

- Pairs that contain too long words. The length threshold for deciding whether the word is too long is 25 characters.

For the alignment perspective, we use *fast_align* (Dyer et al., 2013) to acquire the alignment scores from source to target and vice versa, then we average the scores for each pair to calculate a data pair's sentence-level alignment score. If a sentence pair's sentence-level alignment score is lower than -15, it will be expelled from the final dataset.

Having purified the corpus, we generally try to boost our systems using the following techniques, step by step:

1. Back-translation and forward-translation. Using the trained models to translate big volume, monolingual corpus from the target side to source side (i.e. back-translation (Sennrich et al., 2016a)) has been proved a very successful method in the past practices. In our experiments we can also see a general improvement brought by this technique, but it is not always the case. We also tried sampling based back-translation proposed in (Edunov et al., 2018), and this is effective only in certain cases as well. Furthermore, we found translating from the monolingal corpus from source language can also bring gains for the models (consistent with the phenomenon depicted in (Burlot and Yvon, 2018)), but in this situation arg-max based beam search should always be applied. We also followed (Hoang et al., 2018) to iteratively back-translate and forward-translate the corpus for several times.

2. Fine-tune. Adding too many synthetic parallel data generated by machine translation models could potentially modify the latent data distribution, and in some tasks the provided monolingual dataset has a small difference from the required domain (news), so after having trained models from the mixture of the original parallel corpus and the synthetic dataset, we continue fine-tune our models on the original parallel datasets only. Besides, for some low-resource tasks (such as tasks on Pashto and Khmer), even the official training datasets have relatively lower qualities, therefore only using training dataset to fine-tune is still not enough. For these tasks, we took one more step to fine-tune the models on the official released validation set, and we can always see a further improvement.

3. Ensemble. We generally train and fine-tune several different models and compose them into an ensemble model for a better result.

4. Reranking. With the ensemble model in the hand, we usually generate k-best candidates and use different scorers to score them. Scorers can be divided into three groups: **forward scorers** are just another ensemble models composed by the forward translation models (models translate the source language to the target language). Suppose we have trained 6 base forward models, typically we compose all of them together to form a big ensemble model for generating final results (this model is also used as a scorer), and then additionally enumerate all the 5-combinations of them to get another $\binom{6}{5} = 5$ scorers. Sometimes we furthermore enumerate all the 4-combination to get $\binom{6}{4} = 15$ more scorers for better reranking. **backward scorers** are ensemble models that actually back-translation models (models translate the target language to the source language), and **language models** are ensemble language models of target language. For each group of the scorers, we may use the left-to-right (l2r) models or right-to-left (r2l) models. For the latter form, we reverse the words orders for both source sentences and target sentences and train the models. The scores

generated by those scorers are used as features by the reranking model. For reranking, we mostly applied K-Batched MIRA (Cherry and Foster, 2012) or noisy channel (Yee et al., 2019).

3 Experiments Details

In this section we demonstrate our experiments details for each direction. For brevity we will ignore the same preprocessing and techniques we introduced in the previous section, mainly focus on how the techniques boosted the systems, and some other unique observations we found during the experiments.

In the text we will sometimes use ISO-639-1 two-letter codes for each language for short. Mapping between the abbreviations and full names can be found in Table 1. For example, when talking about the English → Chinese task, we may write `EnZh` for short, capitalizing the first letter of the ISO-639-1 codes for both source languages and target languages. For the direction pairs that involve English, sometimes we use the non-English language to indicate the whole pair, e.g. "Russian tasks" is used to indict the English ↔ Russian bidirectional task. As this report is in the news task scope, we sometimes use "task" as a synonym of "direction", e.g. "`EnZh` task" means the direction that translates English to Chinese.

By default, for every sub-task we combine all the official provided parallel corpora into a big dataset then clean it, use the cleaned corpus to train our baseline models. We strictly followed the requirement of the contest to use official released datasets only, so the systems we built are all constrained systems. If not mentioned, all of our baseline models are trained on the parallel corpus only, and all the scores reported are calculated by sacreBLEU (Post, 2018) based on the results which has been removed BPE symbols, detruecased and detokenized. We always apply BPE subwords (Sennrich et al., 2016b) on the corpora, usually train Transformer-Big models and tie the input and output matrices of the decoder. For all the tasks, we used Adam optimizer (Kingma and Ba, 2014). All the main systems (i.e. submitted results) are generated by the model listed in the **last** row of the corresponding table in each task.

Language Name	**ISO-639-1 Code**
Chinese	`zh`
Czech	`cs`
English	`en`
French	`fr`
German	`de`
Inuktitut	`iu`
Japanese	`ja`
Khmer	`km`
Pashto	`ps`
Polish	`pl`
Russian	`ru`
Tamil	`ta`

Table 1: ISO-639-1 codes for languages appear in news task of WMT20

3.1 English ↔ Chinese

3.1.1 Data Preprocessing

Compared from the other languages in the shared task, especially the languages which use alphabetical writing systems, Chinese has three typical characteristics, leading to three extra preprocessing steps we introduce below:

1. Chinese has two different writing systems: simplified Chinese and traditional Chinese. Following the statistical information mined from the original parallel corpus, we converted all traditional Chinese characters to their simplified counterparts.

2. Some websites use GB2312 to encode texts, therefore could convert Latin letters, digit characters and some other punctuation marks into *full width* form. Besides of some particular punctuation marks (full stops, commas, question marks and exclamation marks), we converted all the other symbols to half width form.

3. Chinese does not have explicit words boundaries, all the characters in the same clause are connected together. We used *pkuseg* (Luo et al., 2019) to segment words from the text.

It should be noted that Japanese also has these three features, so the same process is also applied in the English ↔ Japanese systems.

For data filtering stage, besides the heuristic rules we demonstrated in the previous section, we also compare the count of numbers and punctuation marks between source side and target side. If the difference on number counts is greater than 3 or the difference on punctuation marks counts is greater than 5, the sentence pairs will also be removed.

3.1.2 Training

We combined the Chinese corpus and English corpus together to train BPE. The total BPE operation merge counts is 36K. After learning BPE operations, we built vocabularies for each language separately. The final vocabulary size for Chinese is 42K and for English is 23K. The model architecture for both directions are all Transformer-big. For `ZhEn` task, we tried different hyperparameters to train several models for getting ensemble model: learning rates ranged from 0.0003 to 0.0008, warmup steps fixed at 16,000, dropout ranged from 0.2 to 0.3. For `EnZh` task, the hyperparameters are all fixed (but tried different random seeds): learning rate was 0.0003, warmup steps was 15,000, feed forward network dimension was 15,000.

Entity substitution is experimented in the `ZhEn` system. We use *StanfordNLP* (Qi et al., 2018) to do the NER from parallel corpus and Chinese monolingual datasets (Because in Chinese monolingual datasets an annotation usually follows a foreign name). After having extracted all the entities, we didn't use alignment information to build the mapping between Chinese entities and English entities, but constructed such relationship just according to co-occurrence frequency information: suppose an entity “北京” occurs 50 times totally in the Chinese corpus from 20 sentences, and in the corresponding 20 English sentences “Beijing” occurs 51 times, “Shanghai” occurs 10 times, then we believe “北京” can be translated to “Beijing” but not “Shanghai”. With the entity mapping rules, we then replace the entities in the sentence pairs by different tags `<tag1>`, `<tag2>` ... and train models. In the inference time, model generates results with those tags, and we take another post-edit stage to recover the entities, using the mapping rules as lookup tables.

Table 2 shows our systems for `ZhEn` task, and 3 shows our systems for `EnZh` task. For `ZhEn`, we back-translated 20M NewsCrawl and 17M NewsDiscussion monolingual datasets from English to Chinese, and forward-translated 13M Chinese monolingual dataset to English (including XMU, LDC, etc.).

System	BLEU	Improvement
Baseline	28.8	-/-
+ Back-translation	29.8	+1.0/+1.0
+ Forward-translation	34.5	+5.7/+4.7
+ Entity substitution	35.2	+6.4/+0.7
+ Fine-tuned by `newstest2017`	36.7	+7.9/+1.5
+ Ensemble & reranking	38.3	+9.5/+1.6

Table 2: Overview of our WMT20 Chinese → English systems. In the “Improvement” column we report two improvement amounts, the first one is the improvement amount compared with the baseline model (absolute improvement), and the last one is got from comparing with the previous step (relative improvement). Scorers for reranking are composed by 3 forward left-to-right (l2r) models, 3 forward right-to-left (r2l) models, 3 backward r2l models and 2 l2r Transformer language models.

System	BLEU	Improvement
Baseline	38.6	-/-
+ Back-translation (A)	39.1	+0.5/+0.5
+ Fine-tuned by parallel corpus	40.6	+2.0/+1.5
+ Fine-tuned by `newstest2017`	41.3	+2.7/+0.7
+ Forward-translation (B)	41.9	+3.3/+2.8
+ Ensemble	42.7	+4.1/+0.8
+ Reranking	43.2	+4.6/+0.5

Table 3: Overview of our WMT20 English → Chinese systems. BLEU scores are character-level. Model trained by adding forward-translation data (system B) is directly compared with the one trained by adding back-translation data only (system A). The two phases fine-tune, which is effective for the system A, has no obvious impact on system B

3.2 English ↔ Czech

3.2.1 Data Preprocessing

The officially released English ↔ Czech dataset has a different format from the other sub-tasks. The dataset, which is called CzEng 2.0 (Kocmi et al., 2020), contains not only parallel sentence pairs, but also the data source and three scores: alignment score calculated by dual conditional cross-entropy filtering (Junczys-Dowmunt, 2018), and language scores to show of how confident the source is Czech and the target is English. This extra information can further help us to filter the corpus.

Having noticed that both CsEn and EnCs tasks would be evaluated on long, document-level news datasets, and the CzEng dataset contains some document information, we first analyzed the data sources given in the dataset, to determine which of them are near to the destination domain, and which are far away. The data sources were observed from four aspects: 1. Are the sentences more colloquial or more formal? 2. How well the data is aligned? 3. Can the sentences form a paragraph? 4. Is the corpus also in the news domain?

With the features of the given data sources, we first set a hard condition to check whether a given sentence pair could be kept, then set different probabilities to randomly drop some pairs from certain data sources. Constrained by the paper length we cannot list all of the rules for all the data sources here, but we can take some examples. For the data of which the source is *news*, we kept all of them; at the other extreme, for the *commoncrawl* data, we first removed all the data pairs of which the alignment scores are below than 0.25, or the probabilities of the source sentences belonging to Czech are less than 0.9, then we removed 40% of the remained data randomly.

As the original dataset contains some paragraph information, we concatenated all the sentences that were originally in the same paragraph with a delimiter "|||" (for the sentences that come from the data sources of *subtitles*, *subtitleE* and *subtitleM*, we didn't concatenate them). After the initial filtering, we kept 24.24 million data pairs (If we add in the czeng-test data, the total volume is 24.44 million pairs). The kept data were then processed and filtered by the pipeline presented in the previous section, and we finally got 14.4 million pairs. Detailed preprocessing information can be found in Table 4.

Step	# Sentence pairs kept	Retention rate
Initial filtering	24.44 M	-
Deduplication	17.3 M	70.75%
Heuristic filtering	14.42 M	83.41%
Bad characters filtering	14.40 M	99.84%

Table 4: Preprocessing of the CzEng dataset. Official provided dataset contains alignment information so we didn't calculate alignment scores again, directly reused official information in the initial filtering step. In the "bad characters filtering" step, we printed a character frequency list from the dataset and set a threshold, removed all data pairs that contain irregular characters whose frequencies are lower than the threshold.

System	Score
full-doc	25.7
short-doc	26.3
no-doc	27.0

Table 5: Document-level model training experiments on the EnCs task. Scores are reported on newstest2019 dataset

3.2.2 Model Training

As the evaluation for the En ↔ Cs tasks would be document-level, we first experimented to see if training a model on a dataset which contains many very long sentences can generate better translations for whole documents. We prepared the datasets in three different ways: 1. Concatenating all sentences that belong to the same document (as indicated in the original data sources), noted as "full-doc"; 2. Concatenating three consecutive sentences together, and select the middle one as the final result from the generated translation, noted as "short-doc"; 3. No special preprocessing, one line contains one sentence, noted as "no-doc". The experiments results are shown in Table 5.

From the results we can find that no extra document related preprocessing is the best preprocessing, so we continued our improvement based on the dataset which does not contain document-level information. We first trained two models based on the full CzEng 2.0 dataset (including all the official translated data). Models are all trained using Transformer-Big architecture with norm clipping set to 0.1, dropout set to 0.3, gradient update frequency set to 8, maximum tokens in a batch set to 6000. Warmup steps and learning rate varied from different experiments, the most common combination is warmup steps set to 16,000 and learning rate set to 0.001. During decoding the beam size is 5 and length penalty is 2.5 for CsEn, 2 for

Direction	Dataset	# Data pairs	Score
CsEn	All official released data	122 Million	34.0
CsEn	All official released data + 31M full sampling back-translated data	153 Million	34.1
CsEn	30M data sampled from official released data + 31M full sampling back-translated data	61.2 Million	34.2
EnCs	All official released data	122 Million	28.6
EnCs	All official released data + 28M full sampling back-translated data	150 Million	29.0

Table 6: Models prepared for the final back-translation and forward-translation. Czech monolingual datasets are the combination of all officially provided Newscrawl datasets, English monolingual datasets are sampled from Newscrawl 2019.

EnCs. The score of the CsEn model on offline test set (newstest2018) is 34.0 and the EnCs model on validation set (newstest2019) is 28.6. We use these two models back-translated and forward-translated several data, mixed our synthetic dataset with the original official whole datasets together, and trained several models. Models which have the best performances are selected for the final back-translation and forward-translation, which are listed in Table 6.

We composed the models shown above as two ensemble models, one for each direction, and did another round of back-translation and forward-translation again. For the EnCs task, we prepared two different final datasets as below. Two datasets are all generated by randomness-based back-translation, the difference is the full sampling one sample output words in the full vocabulary, whilst the top-k one restricts the sampling pool in the words that are listed in the top-k highest probabilities for each step:

- **Top-k sampling based dataset**, consists of 24 million data pairs from the original parallel corpus, 54 million officially provided forward-translated corpus (translated from English monolingual corpus), 50 million top-10 sampling back-translated corpus, and 15 million forward-translated corpus generated by our own ensemble model.

- **Full sampling based dataset**, consists of 24 million data pairs from the original parallel corpus, 54 million officially provided forward-translated corpus (translated from English monolingual corpus), 15 million forward-translated corpus generated by our own ensemble model, 31 million "old" full sampling back-translated data used in Table 6, and 36.7 million "new" full sampling back-translated data generated by ensemble model.

System	BLEU	Improvement
Baseline (parallel data only)	27.0	-/-
+ Officially provided synthetic data	28.6	+1.6/+1.6
+ Full sampling based back-translated data	29.0	+2.0/+0.4
+ Ensemble	29.2	+2.2/+0.2
+ FDA fine-tune	29.7	+2.7/+0.5
+ Fine-tune by Newstest2018 & reranking	30.5	+3.5/+0.8

Table 7: Overview of our WMT20 English → Czech systems. Scorers for reranking are composed by 16 forward left-to-right (l2r) models, 3 forward right-to-left (r2l) models, 3 backward r2l models and 3 l2r Transformer language models. We re-learned BPE after adding in the officially provided synthetic data and fixed it for the following steps. The BPE is learned separately and the merge operations count is 36K.

> The 36.7 million "new" back-translated data are generated after an extra cleaning step: As we observed the results generated by full-sampling back-translation sometimes contain very bad sentences, we check how many steps the decoder scores below -10 when decoding for a given input. If 20% of the step scores for a given sentence are below -10, then we discard the sentence pair.

We found the models trained by top-k sampling based dataset are generally worse than those trained by full sampling based dataset, therefore selected one top-k sampling based model and three full sampling based model to form the final ensemble model for decoding the test data. For the CsEn task, The final dataset is composed by 24 million original parallel data pairs, 24 million ensemble knowledge distillation data pairs, 50 million top-k sampling back-translated pairs, 10 million argmax beam search back-translated pairs, and 17 million forward-translated pairs. We trained 4 models using different learning rate (varied from 0.0008 to 0.0015) on this dataset, and fine-tuned them using original parallel dataset (fine-tuning on EnCs models does not bring any gains). The fine-tuned models are used for the final ensemble model. We also applied FDA algorithm (Biçici and Yuret, 2011) on the parallel dataset, picked out 5 million sentence pairs that are similar to the test set and fine-tuned on this small dataset.

The overview of our EnCs system is listed in Table 7, and CsEn system is listed in Table 8

3.3 English ↔ German

For En ↔ De tasks, we generally followed the process depicted in Section 2, cleaned 46.8 million data pairs and kept 30.6 million. For data

System	BLEU	Improvement
Baseline	31.9	-/-
+ Officially provided synthetic data	34.0	+2.1/+2.1
+ Full sampling based back-translated data	34.1	+2.2/+0.1
+ Original parallel data fine-tune	34.8	+2.9/+0.7
+ Ensemble	35.3	+3.4/+0.5
+ FDA fine-tune	35.5	+3.6/+0.2
+ Reranking	35.9	+4.0/+0.4

Table 8: Overview of our WMT20 Czech → English systems. Scorers for reranking are composed by 16 forward left-to-right (l2r) models, 3 forward right-to-left (r2l) models, 3 backward r2l models, 3 l2r Transformer language models and 1 all lower-cased Transformer language model which does not apply BPE on the training dataset.

System	DeEn BLEU	EnDe BLEU
Baseline	40.7 (-/-)	42.6 (-/-)
+ KD	-	44.9 (+2.3/+2.3)
+ Fine-tune on parallel corpus	-	45.3 (+2.7/+0.4)
+ Ensemble	41.9 (+1.2/+1.2)	45.9 (+3.3/+0.6)
+ Reranking	42.2 (+1.5/+0.3)	46.5 (+3.9/+0.6)

Table 9: Overview of our WMT20 German ↔ English systems. Reranking follows noisy-channel reranking (Yee et al., 2019). BLEU scores are reported on newstest2019. We learned BPE jointly for both tasks, merge operation is 32K. Learning rate for training is 0.001 and warmup steps is 4000

preprocessing, we removed sentence pairs that contain too many punctuation marks, and too many [^A-Za-z] characters. In both directions we found neither back-translation nor forward-translation could yield any gains. In the EnDe we found ensemble knowledge distillation (Freitag et al., 2017) could improve the effect but in the DeEn task it did not help. The overview of our En ↔ De system is listed in Table 9.

3.4 English ↔ Inuktitut

We just adapted the official preprocessing script in the syllabic form to process the corpus. BPE was learned independently and the merge operations count is 16K. The overview of our En ↔ Iu system is listed in Table 10.

System	EnIu BLEU	IuEn BLEU
Baseline	23.7 (-/-)	40.0 (-/-)
+ Back-translation	23.8 (+0.1/+0.1)	40.5 (+0.5/+0.5)
+ Knowledge distillation	-	41.3 (+1.3/+0.8)
+ Ensemble	24.3 (+0.6/+0.5)	41.9 (+1.9/+0.6)
+ Reranking	-	43.7 (+3.7/+1.8)

Table 10: Overview of our WMT20 English ↔ Inuktitut systems. Scores are reported on the official validation set

System	JaEn BLEU	EnJa BLEU
Baseline	22.0 (-/-)	37.0 (-/-)
+ Back-translation	24.5 (+2.5/+2.5)	41.4 (+4.4/+4.4)
+ Knowledge distillation	25.1 (+3.1/+0.6)	41.4 (+4.4/+0.0)
+ Ensemble	25.7 (+3.7/+0.6)	42.1 (+5.1/+0.7)
+ Reranking	26.1 (4.1/+0.4)	42.5 (+5.5/+0.4)

Table 11: Overview of our WMT20 English ↔ Japanese systems. Reranking follows noisy-channel reranking (Yee et al., 2019). BLEU scores are reported on the offline official validation set, for EnJa, we report the character-level score. We trained BPE separately for both tasks, merge operations is 32K. Learning rate for training is 0.0003 and warmup steps is 15000. We tried two different feed forward network dimensions, 4096 and 15000, and found no big differences

3.5 English ↔ Japanese

Our En ↔ Ja systems generally follow our En ↔ Zh systems depicted before, the difference was the upper bound of sentence length limit was set to 180 words, and we also set the lower bound to 3. For Japanese word segmentation we used *mecab* [2]. We cleaned 17.64 million parallel pairs and 13.7 million left. For back-translation, we used 16 million Japanese monolingual data and 13 million English monolingual data. The overview of our En ↔ Ja system is listed in Table 11

We tried to fine-tune the models using original parallel dataset, but didn't see any gain. After the test dataset was released, we applied FDA algorithm and extracted 5000 sentences from the training dataset which are the most similar to the test data. These sentences are mixed with the original validation dataset together, then 500 sentences are split out as a new validation set, the rest were used to fine-tune the models. This step improved our EnJa system by 1.3 BLEU and for JaEn it is 0.4 BLEU. However, as validation dataset changed and the scores on the new validation dataset were extremely high, this step is not listed in the Table 11.

3.6 English ↔ Khmer

For the Khmer tasks (and some other tasks in the following), The data preprocessing stages are slightly different from the way we depicted in the second section, stricter in the filtering part, which would remove the sentence pair if...

1. It is a duplicated example

2. The source or target side is empty

[2] https://taku910.github.io/mecab/

3. It contains urls
4. It has words that contain more than 4 consecutive repeated characters
5. It has unpaired quotation marks or parentheses (not applicable for Khmer tasks, but applied in the other tasks shown later)
6. The punctuation marks between the source and the target cannot be matched (not applicable for Khmer tasks, but applied in the other tasks shown later)
7. The length ratio between the source and target is greater than 2.0 or less than 0.5 (for Khmer is between 0.33 and 3)
8. More than half of the tokens are not from the indicated language. We designed a regular expression (noted as regex for short) for each language according to its alphabet, if the word failed to pass the regex, we say it is not from the given language. For example, the regex for English is `[a-zA-Z'-]+`

The maximum sentence length we allowed is also set to 200 words.

Similar to Chinese and Japanese, Khmer does not mark the words boundaries neither, so we used *SEANLP* [3] to do the Khmer word segmentation. After the cleaning, the 4.46 million pairs of sentences had 351K lines left.

It should be noted that the writing system of Khmer, Khmer script, is an *abugida*, means vowels do not have independent symbols, but are stuck after/above/below/in front of the consonants they follow. Roughly, the minimal meaningful unit of Khmer is called Khmer Character Cluster (KCC for short) (Huor et al., 2004), which should be regarded as a whole but actually contains several characters. Original BPE method would break KCC, but this is not what we expect, so we made some modification to keep it (the segmentation tool we used also considered this language feature). We combined Khmer corpus and English to train BPE together, the BPE merge operations count is 8K.

To train the model, we tried different learning rate ranged from 0.0001 to 0.0004, and different warmup steps from 2,000 to 32,000. The overview of our `Km` ↔ `En` system is listed in Table 12. Baseline model is trained by Transformer-mini (4-heads Transformer composed by 4 layers, embedding dimension set to 256, feed forward network dimension set to 1024), learning rate ranged from 0.0008 to 0.001, warmup steps fixed at 40,000. For back-translation, we used all officially provided Khmer monolingual data, and 27 million sentences for English from NewsCrawl 2019 and NewsCommentary 2019.

System	`KmEn` **BLEU**	`EnKm` **BLEU**
Baseline	5.7 (-/-)	2.38 (-/-)
+ Back-translation	13.0 (+7.3/+7.3)	10.15 (+7.77/+7.77)
+ Ensemble	13.6 (+7.9/+0.6)	10.56 (+8.18/+0.41)

Table 12: Overview of our WMT20 Khmer ↔ English systems. We didn't try fine-tune and reranking for these two tasks. BLEU scores are reported on the official offline validation set, reported on the word-level (different from the online character-level evaluation). for `EnKm`, the score is calculated by *multi-bleu*.

System	`EnPs` **BLEU**	`PsEn` **BLEU**
Baseline	6.0 (-/-)	12.3 (-/-)
+ Back-translation	10.7 (+4.7/+4.7)	14.5 (+2.2/+2.2)
+ Knowledge distillation	10.7 (+4.7/+0.0)	14.8 (+2.5/+0.3)
+ Ensemble	11.0 (+5.0/+0.3)	15.4 (+3.1/+0.6)

Table 13: Overview of our WMT20 English ↔ Pashto systems. BLEU scores are reported on the offline official validation set

3.7 English ↔ Pashto

Our Pashto systems used the similar process we described in the Japanese tasks. We cleaned the 1 million original parallel dataset and kept 700K pairs. BPE was jointly learned and the merge operations count is 10000, but the source language does not share vocabulary with the target. When training the models, the learning rate was set to 9×10^{-4} and warmup steps was 6000. The overview of our `En` ↔ `Ps` system is listed in Table 13

As what we did in the Japanese tasks, we selected 10000 sentence pairs from the training dataset according to the test data using FDA, mixed them with official validation set and devtest set to fine-tune our models for 5 epoch, then reranked the generated candidates. This improved our `EnPs` system by 1.6 BLEU and for `PsEn` the gain is 3.5.

3.8 English ↔ Polish

For `En` ↔ `Pl` tasks, we generally followed the process depicted in `En` ↔ `De` tasks, cleaned 10.3 million data pairs and kept 5.265 million. The overview of our `En` ↔ `Pl` system is listed in Table

[3] https://github.com/zhaoshiyu/SEANLP

System	EnPl BLEU	PlEn BLEU
Baseline	24.9 (-/-)	30.0 (-/-)
+ Back-translation	28.2 (+3.3/+3.3)	33.0 (+3.0/+3.0)
+ Knowledge distillation	28.8 (+3.9/+0.6)	34.6 (+4.6/+1.6)
+ Ensemble	29.9 (+5.0/+1.1)	35.1 (+5.1/+0.5)
+ Reranking	30.0 (+5.1/+0.1)	35.5 (+5.5/+0.4)

Table 14: Overview of our WMT20 English ↔ Polish systems. Reranking follows noisy-channel reranking. BLEU scores are reported on official released validation dataset.

14. Training methods listed are generally the same as what we did for `En` ↔ `De`, the only difference is we separately trained BPE for the two languages (so obviously they no longer share the vocabulary), but BPE merge operations count is still set to 32K.

3.9 English ↔ Russian

The data preprocessing for `En` ↔ `Ru` tasks is the same as demonstrated in the `En` ↔ `Km` part, the only difference is for Russian, our BPE merge operations count is set to 36K. The official released parallel dataset (without official synthetic dataset) is reduced from 43.8 million pairs to 26.5 million after the cleaning. For Russian tasks, we trained the model with some extra rounds of back-translation and knowledge distillation, which are:

- In the first round back-translation, we only used all the official released data including the synthetic part. After training had converged, we continued training on the parallel dataset.
- In the second round back-translation, we added in the back-translated results generated by our models, and continued training again.
- In the knowledge distillation step, we added in the knowledge distillation results on the base of the dataset produced in the previous step. After training had converged, models are continue trained using the mixture of original parallel dataset and the knowledge distillation results.

Full results can be referred to Table 15.

System	EnRu BLEU	RuEn BLEU
Baseline	32.1	38.7 (-/-)
+ Bigger ffn dim	32.6 (+0.5/+0.5)	38.8 (+0.1/+0.1)
+ 1st. round back-translation	32.7 (+0.6/+0.1)	39.0 (+0.3/+0.2)
+ 2nd. round back-translation	33.6 (+1.5/+0.9)	39.6 (+0.9/+0.6)
+ knowledge distillation	34.1 (+2.0/+0.5)	40.4 (+1.7/+0.8)
+ Fine-tune	35.2 (+3.1/+1.1)	40.9 (+2.2/+0.5)
+ Ensemble	35.7 (+3.6/+0.5)	41.3 (+2.6/+0.4)
+ Reranking	35.5 (+3.4/-0.2)	41.7 (+3.0/+0.4)

Table 15: Overview of our WMT20 English ↔ Russian systems. BLEU scores are reported on `newstest2019`. "Bigger ffn dim" means we augmented the dimension of fast forward layer to 8192. In the step "Fine-tune" we fine-tuned our models using the mixture of `newstest2017` and `newstest2018`

System	EnTa BLEU	TaEn BLEU
Baseline	7.6 (-/-)	14.4 (-/-)
+ Back-translation	13.1 (+5.5/+5.5)	26.2 (+11.8/+11.8)
+ Fine-tune*	20.2 (+12.6/+7.1)	31.5 (+17.1/+5.3)
+ Ensemble*	20.4 (+12.8/+0.2)	32.5 (+18.1/+1.0)
+ Reranking*	21.6 (+14.0/+1.2)	32.7 (+18.3/+0.2)

Table 16: Overview of our WMT20 English ↔ Tamil systems. BLEU scores are reported on `newsdev2020`. Configurations can be referred to the Khmer tasks. Steps with extra $*$ marks are evaluated in the tiny 200 lines new validation set.

3.10 English ↔ Tamil

Similar to Khmer, Tamil language also uses abugida. So with the same idea, we need to determine the minimal unit to be separated during BPE training. Here we see syllables as the minimal unit, use *open-tamil* [4] to separate syllables, and use our modified *subword-nmt* to learn BPE separations. Cleaning process is the same as we described in the Khmer tasks, we cleaned all the parallel corpora which contains 660K pairs, and had 450K pairs left. For back-translation, we used all available Tamil monolingual corpus (27 million lines totally) and 16 million English sentences sampled from NewsCrawl 2019 and NewsCommentary 2019. BPE is learned jointly, the merge operations count is 10K. The overview of our `En` ↔ `Ta` system is listed in Table 16. In the fine-tune stage, we randomly kept 200 sentences from the `newsdev2020` as the validation set, and the rest 1,789 sentences are used to fine-tune the model.

3.11 French ↔ German

Our `Fr`↔`De` systems generally followed the steps we described in the Russian tasks, with two differences. The first is that we have only one round back-translation, since for this task pair no official back-translation dataset was released; the second is we didn't continue training using parallel corpus after the model had converged. Following the process described in the Khmer tasks, we cleaned the 13.7 million data pairs and kept 11 million. For

[4] `https://github.com/Ezhil-Language-Foundation/open-tamil`

System	FrDe **BLEU**	DeFr **BLEU**
Baseline	28.9 (-/-)	35.4 (-/-)
+ Back-translation	36.2 (+7.3/+7.3)	36.4 (+1.0/+1.0)
+ knowledge distillation	36.2 (+7.3/+0.0)	36.6 (+1.2/+0.2)
+ Fine-tune	36.3 (+7.4/+0.1)	37.6 (+2.2/+1.0)
+ Ensemble (4 models)	36.7 (+7.8/+0.4)	37.9 (+2.5/+0.3)
+ Reranking	36.8 (+7.9/+0.1)	38.1 (+2.7/+0.2)

Table 17: Overview of our WMT20 French ↔ German systems. BLEU scores are reported on `newstest2019`. In the step "Fine-tune" we fine-tuned our models using `euelections_dev2019`

back-translation, we took 27 million French sentences (combination of NewsCrawl 2017-2019 and News Commentary datasets) and 40 million German sentences (from NewsCrawl 2019 only). We jointly learned BPE for the two langauges, the BPE merge operations count is 32K. We shared the vocabulary among the two languages and tied all embedding layers and output layer in the model. The overview of our `Fr` ↔ `De` system is listed in Table 17.

4 Conclusion

This report described OPPO's submissions to the WMT20 news translation task. We use the similar data preprocess and filtering strategy for all the tasks, contains statistical information based rules and alignment information based rules. We trained Transformer-Big models for all the directions and applied some mature techniques, like back-translation, ensemble model, fine-tune and reranking, they generally all brought gains for the final results. Our final submissions ranked top in 6 directions (English ↔ Czech, English ↔ Russian, French → German and Tamil → English), third in 2 directions (English → German, English → Japanese), and fourth in 2 directions (English → Pashto and and English → Tamil).

References

Ergun Biçici and Deniz Yuret. 2011. Instance selection for machine translation using feature decay algorithms. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 272–283.

Franck Burlot and François Yvon. 2018. Using monolingual data in neural machine translation: a systematic study. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 144–155.

Colin Cherry and George Foster. 2012. Batch tuning strategies for statistical machine translation. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 427–436. Association for Computational Linguistics.

Chris Dyer, Victor Chahuneau, and Noah A Smith. 2013. A simple, fast, and effective reparameterization of ibm model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 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*, pages 489–500.

Markus Freitag, Yaser Al-Onaizan, and Baskaran Sankaran. 2017. Ensemble distillation for neural machine translation. *arXiv preprint arXiv:1702.01802*.

Vu Cong Duy Hoang, Philipp Koehn, Gholamreza Haffari, and Trevor Cohn. 2018. Iterative back-translation for neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 18–24.

Chea Sok Huor, Ros Pich Hemy, and Vann Navy. 2004. Detection and correction of homophonous error word for khmer language. *Ref. No. PANL10n/Admn/RR*.

Marcin Junczys-Dowmunt. 2018. Dual conditional cross-entropy filtering of noisy parallel corpora. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 888–895.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia. Association for Computational Linguistics.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Tom Kocmi, Martin Popel, and Ondřej Bojar. 2020. Announcing czeng 2.0 parallel corpus with over 2 gigawords. *arXiv preprint arXiv:2007.03006*.

Ruixuan Luo, Jingjing Xu, Yi Zhang, Xuancheng Ren, and Xu Sun. 2019. Pkuseg: A toolkit for multi-domain chinese word segmentation. *CoRR*, abs/1906.11455.

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. In *Proceedings of* *the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53.

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

Peng Qi, Timothy Dozat, Yuhao Zhang, and Christopher D Manning. 2018. Universal dependency parsing from scratch. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 160–170.

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.

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.

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.

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 5700–5705.