HW-TSC's Participation in the WMT 2021 Large-Scale Multilingual Translation Task

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

Abstract

This paper presents the submission of Huawei Translation Services Center (HW-TSC) to the WMT 2021 Large-Scale Multilingual Translation Task. We participate in Small Track #2, including 6 languages: Javanese (Jv), Indonesian (Id), Malay (Ms), Tagalog (Tl), Tamil (Ta) and English (En) with 30 directions under the constrained condition. We use Transformer architecture and obtain the best performance via multiple variants with larger parameter sizes. We train a single multilingual model to translate all the 30 directions. We perform detailed pre-processing and filtering on the provided large-scale bilingual and monolingual datasets. Several commonly used strategies are used to train our models, such as Back Translation, Forward Translation, Ensemble Knowledge Distillation, Adapter Fine-tuning. Our model obtains competitive results in the end.

1 Introduction

This paper introduces our submission to the WMT 2021 Large-Scale Multilingual Translation Task. We participate in Small Track #2, including 6 languages: Javanese (Jv), Indonesian (Id), Malay (Ms), Tagalog (Tl), Tamil (Ta) and English (En) with 30 directions. We consider that the officially provided dataset has the acceptable size and quality and therefore only participate in the constrained evaluation. Our method is mainly based on previous works but with fine-grained data cleaning techniques and a multi-step multilingual training strategy.

For each language pair, we perform multi-step data cleaning on the provided dataset and only keep a high-quality subset for training. At the same time, several training strategies are tested in a pipeline, including Backward (Edunov et al., 2018) and Forward (Wu et al., 2019a) Translation, Multilingual Translation (Johnson et al., 2017), Iterative Joint Training (Zhang et al., 2018), Ensemble Knowledge Distillation (Freitag et al., 2017; Li et al., 2019), Adapter Fine-Tuning (Bapna et al., 2019), and Ensemble (Garmash and Monz, 2016).

Based on the task requirements, we train a single multilingual model that translates all 30 directions. We refer to (Johnson et al., 2017) and employ language tags (Wu et al., 2021). By combining multiple strategies, our model achieves considerable quality improvements in all directions.

Section 2 focuses on our data processing strategies while section 3 describes our training techniques, including model architecture and the iterative training strategy, etc. Section 4 explains our experiment settings and training processes and section 5 presents our experiment results.

2 Data

2.1 Data Source

For all language pairs, we follow the constrained data requirements and take full advantage of the bilingual and monolingual training data available. Table 1 lists the data sizes of each language pair before and after filtering.

2.2 Data Pre-processing

We conduct the following steps to pre-process the 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

Proceedings of the Sixth Conference on Machine Translation (WMT), pages 456–463

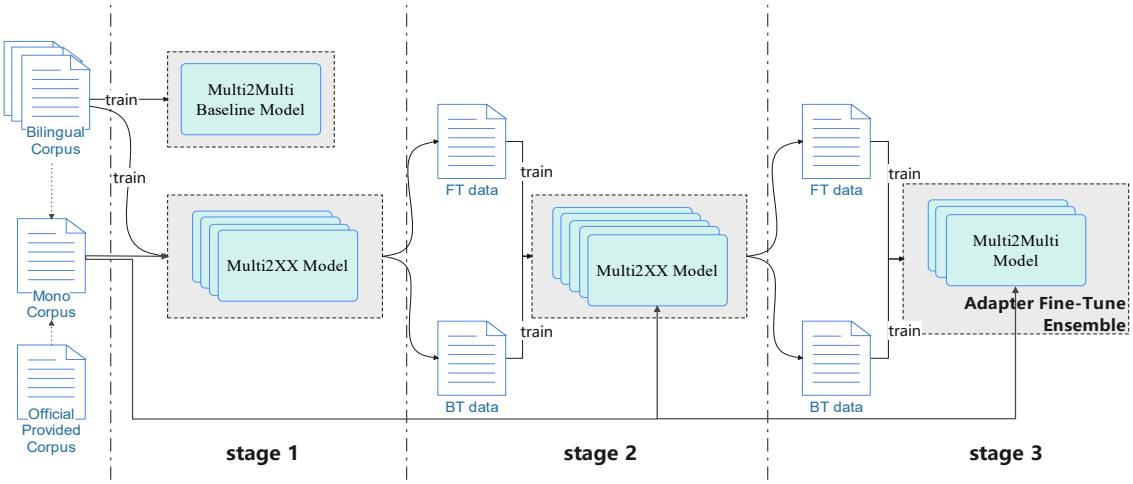

Figure 1: This figure shows the training process for the WMT 2021 Large-Scale Multilingual Translation Task, which consists of three stages. In stage 1, one Multi→Multi model as baseline and five Multi→XX models are trained. In stage 2, the synthetic data by forward and sampling back translation (FTST) is used to train the second round Multi→XX models. In stage 3, second round synthetic FTST data is used to train three Multi→Multi models. Finally, adapter fine-tune and model ensemble are used to enhance the performance.

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 than 3 or lowers than 0.3; and sentences with more than 120 tokens. Based on our experience in the industry, this strategy can reduce the low-level errors in model inference and the problem of missing translations.

- 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.
- Use LaBSE (Feng et al., 2020) to rank and filter the monolingual data.

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

2.3 Data Selection

According to (Arivazhagan et al., 2019), high-resource language pairs may squeeze the living space of low-resource language pairs. In other words, different data sizes across languages may lead to uneven translation quality in a multilingual model. Since we incorporate all 30 directions in one multilingual model, this issues should be addressed. We use temperature sampling strategy (Zoph et al., 2016) with T=5 to over-sample the low-resource language pairs.

We train all 30 directions under the constrained condition. To improve the performance of back-translation, we combine officially provided monolingual data with the monolingual data extracted from corresponding bilingual corpora. Data sizes are listed in Table 1. The detailed bilingual data size after forward translation and sampling back translation (FTST) and over-sampling are listed in Table 3.

3 System Overview

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-Deep (Sun et al., 2019) and consider it as a baseline. The detailed model parameters are as follow: 35-layer encoder, 3-layer decoder, 512 hidden units and a batch size of 4096. We used the Adam optimizer (Kingma and Ba, 2014) with $\beta_1 = 0.9$ and $\beta_2 = 0.98$, and the same warmup and decay strategy for learning rate as (Vaswani et al., 2017), with 4,000 warmup steps. During training, we employ label smoothing with a value of 0.1 (Szegedy et al., 2016). For evaluation, we use beam search with

Language pairs	Raw bi data	Filtered bi data	Mono data
En/Id	54M	16.5M	En: 80M
En/Jv	3M	2.2M	
En/Ms	13.4M	12.1M	
En/Ta	2.1M	1.9M	
En/Tl	13.6M	8.7M	
Id/Jv	0.78M	0.51M	Id: 58M
Id/Ms	4.8M	4.3M	
Id/Ta	0.5M	0.4M	
Id/Tl	2.7M	1.6M	
Jv/Ms	0.43M	0.26M	Jv: 3.8M
Jv/Ta	0.06M	0.037M	
Jv/Tl	0.8M	0.32M	
Ms/Ta	0.37M	0.32M	Ms: 19.7M
Ms/Tl	1.3M	0.8M	Tl: 12.2M
Ta/Tl	0.5M	0.3M	Ta: 5M

Table 1: Bilingual data sizes before and after filtering, and monolingual data used in the task. The monolingual data includes officially provided monolingual data and the mono data extracted from the bilingual corpus of corresponding languages.

a beam size of 4 and length penalty $\alpha = 0.6$ (Wu et al., 2016).

3.2 Data Augmentation

Back-translation (Edunov et al., 2018) is an effective way to enhance translation quality by using monolingual sentences to generate synthetic training parallel data. As described in (Wu et al., 2019b), 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.

We take full advantage of the officially provided monolingual data for data augmentation. In terms of back translation, we adopt top-k sampling for high-resource languages, and adopt beam search for low-resource languages. With regard to forward translation, we translate monolingual data using 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.

3.3 Multilingual Strategy

Johnson et al. (2017) propose a simple solution to use a single neural machine translation 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. According to (Wu et al., 2021), we add "2XX" (XX indicates the target language, e.g. 2id) at the beginning of the source sentence. All languages use a shared vocabulary. We train the hybrid SentencePiece model (Kudo and Richardson, 2018) in conjunction with all 6 languages as the shared word segmentation system for all language pairs. We keep the vocabulary within 40k, including tokens of all 6 languages (En/Id/Jv/Ms/Ta/Tl).

Two mainstream methods about multilingual training are available: two models with XX→Multi and Multi→XX separately and a mono Multi→Multi model. According to (Johnson et al., 2017), Multi→XX performs better than Multi→Multi and XX→Multi in general. Multi→Multi model contains too many languages pairs (30 in this case), so conflicts and confusions may occur among language pairs in different directions. However, due to the requirements of the task, we need to provide a Multi→Multi model that includes all 30 directions. In our experiment, we divide 30 language pairs into five Multi→XX multilingual models as step 1. Than we use five Multi→XX multilingual models to conduct back-translation and train a Multi→Multi model as step 2 and step 3, as shown in Figure 1.

3.4 Iterative Joint Training

Zhang et al. (2018) propose a new iterative joint training method, that is, using monolingual data from both source and target sides to train a source-to-target (forward) model and a target-to-source (backward) model at the same time. The two models generate synthetic data for each other. The advantage of such method is that both of the two models gain improvement after each iteration with the synthetic data provided by the other, and then can generate synthetic data with higher quality. Such training procedure is repeated after the two models converge.

3.5 Language independence Adapter Fine-tuning

Previous works demonstrate that fine-tuning a model with in-domain data could effectively improve the model performance. However, due to limitations of a multilingual translation model, once the model is trained, when fine-tuning one of the language pairs, the performance of others will go worse. Thanks to the finding of Adapter (Bapna et al., 2019), we are able to fine-tune each language pair without impacting the performance of others. In the experiment, we set the adapter size to 512 and fine-tune the model on the bilingual data for each language pair in 30 directions with 3,000 tokens per batch for one epoch.

3.6 Ensemble Knowledge Distillation (EKD)

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 models. Comparing with some soft label distillation methods, the EKD for NMT is relatively straightforward, which can be implemented by training the student models 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 models as the teacher model to translate the FLORES dev set, and use the translation results to further fine-tune models.

3.7 Ensemble

Model ensemble is a widely used technique in previous WMT workshops (Garmash and Monz, 2016), which can improve the performance by combining the predictions of several models at each decoding step. In our work, we ensemble models with different architectures to further improve system performances.

System	FLORES dev	FLORES devtest
baseline M2M	26.9	26.8
FTST1	28.2 (+1.3)	28.1 (+1.3)
FTST2	29.4 (+1.2)	29.6 (+1.5)
Adapter Fine-Tune	30.2 (+0.8)	30.1 (+0.5)
ensemble	30.7 (+0.5)	30.9 (+0.8)
wmt21 final submit	28.6	28.3

Table 2: The experimental results on FLORES dev/devtest, BLEU scores in table are the average of 30 directions.

4 Experiment Settings

4.1 Settings

We use the open-source fairseq (Ott et al., 2019) for training and SentencePieceBLEU to measure system performances. Each model is trained using 8 GPUs. The architectures and main parameters we used are described in section 3.1. Marian (Junczys-Dowmunt et al., 2018) is used for decoding during inference.

4.2 Training Process

We employ iterative training and phase-based data augmentation. Figure 1 shows our training process in details. The specific steps are as follows:

1) Process data using methods described in section 2.2. Train one Multi→Multi model as baseline and five Multi→XX models as forward models and backward models.

2) Generate back translation and forward translation data. Mix the data with parallel training data and train second round five Multi→XX models.

3) Generate back translation and forward translation data using models trained in step 2. Mix data with bilingual training data and train three Multi→Multi models.

4) Average the last eight checkpoints of each model and adapter fine-tune it with bilingual data. Ensemble models to produce the final system.

5 Results and analysis

We use methods described in Section 2.2 for data processing. Model architecture mentioned in Section 3.1 is employed to increase system diversity. On the basis of Multi→Multi baselines model, we use FTST data augmentation to further enhance model performance.

Table 2 lists the results of our experiment on FLORES dev set and devtest set (Goyal et al., 2021). Comparing with the baseline model, the first round FTST Multi→XX models leads to 1.3 BLEU increase on average for the 30 directions. Further, the second round FTST achieves 1.2 BLEU increase on average. We fine-tune the model using bilingual data with adapter and achieve 0.8 BLEU increase on average. Finally, ensemble further leads to 0.5 BLEU increase. When submitting the final results, because of time limits, we only finish round-two FTST. As for model inference, there is a problem with our fairseq architecture, resulting in poor model quality that seriously affects the FTST results. The final model we submitted achieves 28.64 BLEU on FLORES dev and 28.34 BLEU on FLORES devtest. After the submission, we fixed the problem and continued our experiments, eventually achieving 30.7 BLEU on on FLORES dev and 30.9 BLEU on FLORES devtest. The detailed experiment results are listed in Table 4.

In our experiment, due to the inference problem mentioned above, we have not seen much performance improvements. The low quality of model inference leads to poor FT results, which made no contributions to the model. And even worse, it offsets the gain brought by BT results to the model. We also found that the Multi→en model does surpass the Multi→Multi model in quality, which is the same as the results observed by the industry.

6 Conclusion

This paper presents the submissions of HW-TSC to the WMT 2021 Large-Scale Multilingual Translation Task. We perform experiments with a series of pre-processing and training strategies. The effectiveness of each strategy is demonstrated. We finally achieve competitive results.

References

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George Foster, Colin Cherry, et al. 2019. Massively multilingual neural machine translation in the wild: Findings and challenges. *arXiv preprint arXiv:1907.05019*.

Ankur Bapna, Naveen Arivazhagan, and Orhan Firat. 2019. Simple, scalable adaptation for neural machine translation. *arXiv preprint arXiv:1909.08478*.

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*.

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

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.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzman, and Angela Fan. 2021. The flores-101 evaluation benchmark for low-resource and multilingual machine translation. *arXiv preprint arXiv:2106.03193*.

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*.

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

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 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.

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.

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*.

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.

Lijun Wu, Yiren Wang, Yingce Xia, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. 2019a. 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.

Lijun Wu, Yiren Wang, Yingce Xia, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. 2019b. 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.

Liwei Wu, Shanbo Cheng, Mingxuan Wang, and Lei Li. 2021. Language tags matter for zero-shot neural machine translation. *arXiv preprint arXiv:2106.07930*.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Zhirui Zhang, Shujie Liu, Mu Li, Ming Zhou, and Enhong Chen. 2018. Joint training for neural machine translation models with monolingual data. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer learning for low-resource neural machine translation. *arXiv preprint arXiv:1604.02201*.

A Details of Data size and BLEU

Language pairs	Bilingual data	Bi + FTST data	Over-Sampling T=5
En/Id	46M	56M	56M
En/Jv	2.2M	5M	34M
En/Ms	12M	22M	47M
En/Ta	1.9M	12M	41M
En/Tl	8.7M	18.7M	45M
Id/Jv	0.5M	8.1M	38M
Id/Ms	4.3M	24M	47M
Id/Ta	0.4M	10.6M	40M
Id/Tl	1.6M	21.6M	46.8M
Jv/Ms	0.2M	7.8M	38M
Jv/Ta	0.03M	8.9M	39M
Jv/Tl	0.3M	7.9M	38M
Ms/Ta	0.3M	10.5M	40.4M
Ms/Tl	0.8M	20M	46M
Ta/Tl	0.3M	10M	40M

Table 3: Bilingual data sizes before and after FTST, and Bilingual data sizes after over sampling.

Language Pair	Baseline	Final Submit	FTST1	FTST2	Adapter Fine-Tune	Ensemble
En2Id	46.6	49.2	49.4	49.4	49.5	49.8
Id2En	42.8	43.5	43.7	44.1	44.3	44.8
En2Jv	24.6	26.5	26.3	27.1	27.4	28.3
Jv2En	30.6	31.1	30.9	32.4	32.4	33.3
En2Ms	44.4	45.1	44.8	46.7	46.8	47.8
Ms2En	43.6	42.9	42.7	44.3	44.5	46
En2Ta	24.8	24.8	24.6	26.5	26.6	27.2
Ta2En	24.6	24.6	24.3	26.3	26.4	27.3
En2Tl	33.5	35.3	35.2	36.7	37.9	38.6
Tl2En	41.7	42.1	40.9	43.7	43.9	44.7
Id2Jv	19.5	21.3	20.9	23.5	23.8	24.7
Jv2Id	25.9	28.8	28.6	28.9	28.9	29.6
Id2Ms	35.2	36.9	36.7	38	38.8	39.6
Ms2Id	34	38.2	37.8	38.7	38.8	39.7
Id2Ta	20	21.2	20.9	22	22.7	23.2
Ta2Id	17.1	18.8	18.9	19.1	20.3	21
Id2Tl	26.6	28.7	28.4	29.7	30	30.8
Tl2Id	31	33.7	33.9	35	36.2	36.8
Jv2Ms	24.8	26.8	26.9	28.9	29.8	30.2
Ms2Jv	19.7	21.2	21.4	22.5	23.4	23.8
Jv2Ta	13	14.3	13.9	15	16.2	17.5
Ta2Jv	8.6	9.8	10	11.2	12.8	13.1
Jv2Tl	17.6	20.5	19.9	22.3	22.7	23.4
Tl2Jv	15.5	17	17.2	19.4	19.9	20.2
Ms2Ta	20.2	22.1	20.5	24.1	24.3	24.9
Ta2Ms	19.3	20.4	20.5	22.3	22.5	23.2
Ms2Tl	27.7	28	27.6	30.2	30.6	31.5
Tl2Ms	31	32.5	32	33.7	34.1	34.8
Ta2Tl	18.4	20.8	20.1	21.9	23.1	23.7
Tl2Ta	21.9	23.1	23.2	24.9	26.2	27.1
Average	26.8	28.3	28.1	29.6	30.1	30.9

Table 4: BLEU for each direction on FLORES devtest