HW-TSC Systems for WMT22 Very Low Resource Supervised MT Task

Shaojun Li, Yuanchang Luo, Daimeng Wei, Zongyao Li, Hengchao Shang, Xiaoyu Chen, Zhanglin Wu, Jinlong Yang, Zhiqiang Rao, Zhengzhe Yu, Yuhao Xie, Lizhi Lei, Hao Yang, Ying Qin
Huawei Translation Service Center, Beijing, China
{lishaojun18,luoyuanchang,weidaimeng,lizongyao,shanghengchao,chenxiaoyu35,
wuzhanglin2,yangjinlong7,raozhiqiang,yuzhengzhe,xieyuhao2,
leilizhi,yanghao30,qinying}@huawei.com

Abstract

This paper describes the submissions of Huawei translation services center (HW-TSC) to the WMT22 Very Low Resource Supervised MT task. We participate in all 6 supervised tracks including all combinations between Upper/Lower Sorbian (Hsb/Dsb) and German (De). Our systems are build on deep Transformer with a large filter size. We use multilingual transfer with German-Czech (De-Cs) and German-Polish (De-Pl) parallel data. We also utilize regularized dropout (R-Drop), back translation, fine-tuning and ensemble to improve the system performance. According to the official evaluation results on OCELoT[1], our supervised systems for all 6 language directions got the highest BLEU scores among all submissions. Our pre-trained multilingual model for unsupervised De2Dsb and Dsb2De translation also gains the highest BLEU.

1 Introduction

In this paper, we describe our very low resource supervised MT systems for all combinations between Hsb, Dsb and De. We first select a base pre-trained multilingual model and then fine-tune it. As we focus primarily on the supervised task, we only apply our pre-trained multilingual system with zero-shot for unsupervised task submissions.

As show in WMT21 shared task (Libovický and Fraser, 2021), most participants use De-Cs for transfer or combine De-Cs with the low resource pairs to build a multilingual system. Fine-tuning based on a multilingual pre-trained (Fan et al., 2020) model has shown very promising results for low resource tasks. We add De-Pl data and train our multilingual pre-trained model to transfer the low resource pairs.

This paper is structured as follows: we describe our data source and data pre-processing method in section 2. We detail the model structure and method we used in Section 3. We then present the final experiments in Section 4 and Section 5, and finally we conclude our work in Section 6.

[1] https://ocelot-wmt22.mteval.org

2 Dataset

2.1 Data Source

For our base pre-trained multilingual systems, we use all the bilingual data (De-Cs and De-Pl) from the latest version of OPUS. We also sample 20M German monolingual data from news (general) MT task for augmentation. For fine-tuning the systems transfer to our task, we use all the bilingual and monolingual data officially provided without any filtering strategy. We use dev set and test set together for model parameter adjustment and system selection (do not include the blind test data from the previous years).

2.2 Data Pre-processing

For all the data mentioned above, we remove duplicate sentences (Khayrallah and Koehn, 2018; Ott et al., 2018).

For De-Cs and De-Pl, the data pre-processing procedure is as follows:

- Remove sentences with mismatched parentheses and quotation marks.
- Filter out sentences of which punctuation percentage exceeds 0.4.
- Filter out sentences with a character-to-word ratio greater than 12 or less than 1.5.
- Filter out sentences with more than 150 words.
- Apply langid (Joulin et al., 2017, 2016) to filter out sentences in other languages.
- Use fast-align (Dyer et al., 2013) to filter out sentence pairs that are poorly aligned.

Proceedings of the Seventh Conference on Machine Translation (WMT), pages 1098 - 1103

	bilingual					monolingual		
	De-Cs	**De-Pl**	**De-Dsb**	**De-Hsb**	**Dsb-Hsb**	**De**	**Dsb**	**Hsb**
Raw data	77.1M	98.1M	40K	449K	63K	-	220K	1.13M
Processed data	55.9M	66.5M	39K	317K	63K	20M	177K	957K

Table 1: The data sizes of before and after pre-processing in our very low resource supervise MT Task

We sample 3.2M Cs and Pl data from bilingual data and up-sampling 3.2M De, Hsb, Dsb from a combined dataset of all the three languages. We mix the data above and build a joint SentencePiece model (SPM) (Kudo and Richardson, 2018; Kudo, 2018) for word segmentation, with a vocabulary of 40k. We use Moses tokenizer (Koehn et al., 2007) to pre-segment sentences. We also use the combined data to build a joint vocabulary for all of our models. The vocabulary size is slightly larger than SPM vocabulary to cover more tokens, which is set to 41k.

3 System Overview

3.1 Model

Transformer (Vaswani et al., 2017), as the current mainstream architecture of NMT, adopts a fully self-attention mechanism, which can realize algorithm parallelism, speed up model training, and improve model performance. Deep transformer is an variant of Transformer, which increases the number of encoder layers and uses pre-layer-normalization to further improve model performance. Therefore, in all translation tasks, we adopt the following model architecture:

- Deep Transformer (Wei et al., 2021): We refer to the Transformer-big model architecture and decrease the dim for faster training. our Deep Transformer model features pre-layer-normalization, 35-layer encoder, 6-layer decoder, 16-head self-attention, 768-dimension word embedding and 3072-hidden-state.

3.2 Multilingual Transfer

Recent researches have shown that multilingual models outperform their bilingual counterparts, particularly when the number of languages in the system is limited and those languages are related (Lakew et al., 2018). This is mainly due to the capability of the model to learn interlingual knowledge (shared semantic representation between languages) (Johnson et al., 2016) (Ranathunga et al., 2021). Transfer learning using pre-trained multilingual model (Fan et al., 2020) has shown very promising results for low resource tasks. In this task, we first select a multilingual system as the base system, then fine-tune the system with low resource language pairs.

3.3 R-Drop

Dropout (Srivastava et al., 2014) is a powerful and widely used technique for regularizing deep neural networks. Though it can help improve training effectiveness, the randomness introduced by dropouts may lead to inconsistencies between training and inference. R-Drop (Wu et al., 2021) forces the output distributions of different sub models generated by dropout be consistent with each other. Therefore, we use R-Drop to augment the pre-trained multilingual model for each track and reduce inconsistencies between training and inference.

3.4 Back Translation

Back translation (BT) (Edunov et al., 2018) refers to translating the target monolingual data into the source language, and then using the synthetic data to increase the training data size. This method has been proven effective to improve the NMT model performance. We apply sampling(Graça et al., 2019) back-translation for all language directions.

4 Experimental Settings

During the training phase, we use Pytorch-based Fairseq[2] (Ott et al., 2019) open-source framework. Each model is trained using 8-V100 with a batch size of 2048 tokens for each GPU. Dropout was set to 0.1 for pre-train multilingual model, and 0.3 for fine-tuning model. The label smoothing rate (Szegedy et al., 2016) is 0.1. Adam optimizer (Kingma and Ba, 2015) with $\beta 1$=0.9 and $\beta 2$=0.98 is also used. Furthermore, we use reg_label_smoothed_cross_entropy as the loss function and set reg-alpha to 5 when applying R-

[2] https://github.com/facebookresearch/fairseq

System	De2Hsb	Hsb2De
Pre-trained model	1.2	3.6
Bitext finetune	65.6	66.3
Noisied ST	65.8	-
Sampling BT	69.2	67.0
FT+ST	-	67.1
Ensemble	69.4	67.5
WMT22submission	**70.7**	**71.9**

Table 2: Avg. scores on WMT21 dev set, test set and WMT22 dev set for De↔Hsb.

System	De2Dsb	Dsb2De
Pre-trained model	0.9	2.5
Bitext finetune	50.1	55.2
Noisied ST	50.6	-
Sampling BT	58.0	57.8
FT+ST	-	57.9
Ensemble	58.2	58.1
WMT22submission	**73.9**	**62.5**

Table 3: Avg. scores on WMT21 dev set, test set and WMT22 dev set for De↔Dsb.

Drop training strategy. For pre-training the multilingual model, the update frequency, the learning rate and warm-up steps are 4, 5e-4 and 4000 respectively; for fine-tuning the model, the update frequency and the learning rate is 1 and 1e-4 without warm-up. In the evaluation phase, we use Marian[3] (Junczys-Dowmunt et al., 2018) for decoding and then calculate the sacreBLEU[4] (Post, 2018) on the WMT21 dev set, test set and WMT22 dev test to measure the performance of each model.

5 Experimental Result

First of all, we test pre-trained multilingual models without fine-tune and get quiet low scores. Next, we fine-tune all of our pre-trained models with bitext and then select a best one according to the BLEU scores for every task.

5.1 De↔Hsb

Table 2 shows the results of using the selected pre-trained multilingual model to improve the De↔Hsb model performance.

In De2Hsb, we adopt the strategy of noised ST (Imamura and Sumita, 2018) because we have a large amount of German monolingual data. We sample 20M German monolingual for noised ST.

[3] https://github.com/marian-nmt/marian

[4] https://github.com/mjpost/sacreBleu

System	Hsb2Dsb	Dsb2Hsb
Pre-trained model	1.1	1.0
Bitext finetune	62.9	65.3
Multilingual finetune	67.6	72.0
Sampling BT	69.6	74.2
Ensemble	70.0	74.7
WMT22submission	**88.0**	**86.8**

Table 4: Avg. scores on WMT22 dev set for Dsb↔Hsb.

We find that this strategy can bring an additional 0.2 BLEU improvement. At the same time, we use all the Hsb monolingual data (including the Hsb side of Dsb-Hsb) for sampling BT, which brings an increase of 3.4 BLEU. BLEU increases by 0.2 after ensemble.

In Hsb2De, We find that both sampling BT and forward translation + sampling BT (FT+ST) (Wu et al., 2019) can bring certain improvement, but sampling BT outperforms the FT+ST strategy (0.7 BLEU vs 0.1 BLEU). After ensemble, model performance continues improving by 0.4 BLEU.

5.2 De↔Dsb

Table 3 shows the results of using the selected pre-trained multilingual model to improve the De↔Dsb model performance. We follow the same strategy as that of De↔Hsb.

In De2Dsb, we adopt noised ST with the same data as De2Hsb. We use 20M German monolinguals for noised ST. We find that this strategy can bring an additional 0.5 BLEU improvement. At the same time, we use all the Dsb monolingual data (including the Dsb side of Dsb-Hsb) for sampling BT, which brings an improvement of 7.4 BLEU. Finally, BLEU increases by 0.2 after ensemble.

In Dsb2De, Sampling BT and FT+ST can bring certain improvement (2.6 BLEU vs 0.1 BLEU). After ensemble, model performance continues improving by 0.2 BLEU.

5.3 Hsb↔Dsb

Table 4 shows the results of using the selected pre-trained multilingual model to improve the Hsb↔Dsb model performance.

Regarding the Hsb2Dsb task, we first fine-tune the many-to-many pre-trained model with bitext, and then combine the De2Hsb, De2Dsb, Hsb2De, Dsb2De multilingual to continue fine-tuning both Hsb2Dsb and Dsb2Hsb models. This strategy gets improvements of 5+ BLEU. Then, we perform one

Pre-trained model	De2Hsb	De2Dsb	Hsb2De	Dsb2De	Hsb2Dsb	Dsb2Hsb
De2Cs	64.5	49.1	-	-	61.0	63.2
Cs2De	-	-	64.7	51.9	-	-
one-to-many	**64.6**	**49.4**	-	-	61.8	63.6
many-to-one	-	-	**65.3**	**54.0**	-	-
many-to-many	64.6	49.2	64.9	53.0	**62.0**	**64.3**

Table 5: Avg. scores on WMT21 dev set,test set and WMT22 dev set with different pre-trained models. **De2Cs**: pre-train with De2Cs bilingual data; **Cs2De**: pre-train with Cs2De bilingual data; **one-to-many**: pre-train with De2Cs and De2Pl bilingual data; **many-to-one**: pre-train with Cs2De and Pl2De bilingual data; **many-to-many**: pre-train with Cs2De, De2Cs, Pl2De and De2Pl bilingual data.

System	De2Hsb	De2Dsb	Hsb2De	Dsb2De	Hsb2Dsb	Dsb2Hsb
w/o R-drop	64.6	49.4	65.3	54.0	62.0	64.3
w/ R-drop	**65.6**	**50.1**	**66.3**	**55.2**	**62.9**	**65.3**

Table 6: Avg. scores of WMT21 dev set, test set and WMT22 dev set for each track without or with R-drop.

round of sampling BT for optimization.

After we ensemble the latter two models, the model performances significantly increase by 7.1 and 9.4 BLEU respectively when comparing with the base many-to-many fine-tuning model, which also proves the advantages of the multilingual model in low-resource tasks.

5.4 Unsupervised Submission

We conduct an unsupervised experiment with our many-to-one and one-to-many pre-trained model. For Hsb2De and Dsb2De, we add tags to the Hsb/Dsb monolingual data and get the German result from many-to-one model for zero-shot. Then we fine-tune the best one-to-many model with the zero-shooting translations, and get the Hsb2De, Dsb2De models, which obtains 11.5 BLEU on the Hsb2De track and 13.5 BLEU on the Dsb2DE track. Based on the two base models, we continue conducting a round of BT with 2M German monolingual data, and then train the De2Hsb and De2Dsb models for submission. The BLEU scores are 10.4 (De2Hsb) and 9.0 (De2Dsb). As we have not invested much efforts in the unsupervised task, more experiments need to be done in the feature.

6 Analysis

6.1 Pre-trained Model

Due to the availability of large amount of De-Cs and De-Pl data and the similarities between Hsb/Dsb and Cs/Pl, we pre-train several multilingual models with different strategies, and then fine-tune with very low resource bilingual data. We choose the best strategy for every language direction for further fine-tuning. Specifically, we design three pre-trained multilingual models: a one-to-many model trained with the De2Cs and De2Pl bilingual data, a many-to-one model trained with the Cs2De and Pl2De bilingual data, and a many-to-many model trained with the Cs2De, De2Cs, Pl2De and De2Pl bilingual data. For each corpus we use a different tag to differentiate. Furthermore, we train a De-Cs model as done by last year's participants, in order to get better comparison results.

Table 5 shows that data selection for the pre-trained model is closely related to the task requirements. For tasks of translating other languages into De, the many-to-one model trained with Cs2De and Pl2De corpora performs the best. For tasks of De2Hsb/Dsb, the one-to-many model trained with De2Cs and De2Pl corpora works the best. In general, the many-to-many model is more suitable for the Hsb2Dsb and Dsb2Hsb task, because the many-to-many model is trained with Cs and Pl data at both the source and target sides. With large amount of Cs/Pl data for transfer, both the encoding and decoding layers can benefit transfer for the Hsb2Dsb and Dsb2Hsb embeddings. Multilingual pre-trained models have better performance than bilingual pre-trained models in all directions. Besides, the unsupervised results also show the transfer capability of our pre-trained multilingual models.

6.2 The Effect of R-drop

Considering the limited sources and the large size of the multilingual model, we try the R-drop strat-

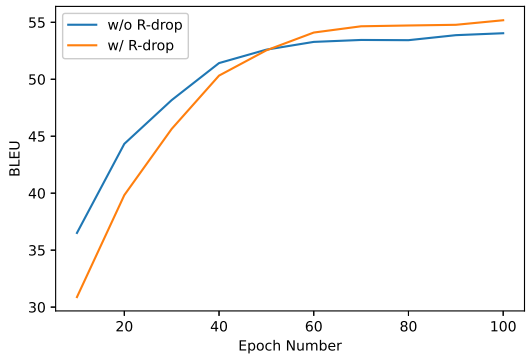

Figure 1: BLEU curves along with or without R-drop

egy to see whether the R-drop strategy is effective on multilingual models. Based on the optimal multilingual model for each task selected in the previous step, we compared whether the use of the R-drop strategy in the training process can lead to further improvements. It can be seen from Table 6 that after using the R-drop strategy for training, the BLEU of each track further improves by at least 1 point, indicating that R-drop does have a good effect in low-resource scenarios with large models. Therefore, we adopt the R-drop strategy for further training. In addition, Figure 1 is a graph depicting the BLEU convergence curves on the Dsb2De track. From the figure we can find that when the BLEU value increases by using R-drop, the convergence time also increases by about ten epochs.

7 Conclusion

This paper presents our translation systems for the WMT22 very low resource supervised MT task. During the experiment, We use multilinguals to build our base translation system, and then use forward translation and back translation methods to expand the size of training data for a better translation system. We also adopt test set fine-tuning and ensemble to further improve the system performance. Finally, according to the official evaluation results on OCELoT, our submission achieves the highest BLEU scores in all 6 language directions in the supervised task, and our submission of De2Dsb and Dsb2De also gains the highest BLEU in unsupervised task.

References

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.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Edouard Grave, Michael Auli, and Armand Joulin. 2020. Beyond english-centric multilingual machine translation. *arXiv: Computation and Language*.

Miguel Graça, Yunsu Kim, Julian Schamper, Shahram Khadivi, and Hermann Ney. 2019. Generalizing back-translation in neural machine translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 45–52.

Kenji Imamura and Eiichiro Sumita. 2018. Nict self-training approach to neural machine translation at nmt-2018. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 110–115.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda B. Viégas, Martin Wattenberg, Greg S. Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Herve Jegou, and Tomas Mikolov. 2016. Fasttext. zip: Compressing text classification models.

Armand Joulin, Édouard Grave, Piotr Bojanowski, and Tomáš Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431.

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++. In *ACL (4)*.

Huda Khayrallah and Philipp Koehn. 2018. On the impact of various types of noise on neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 74–83.

Diederik P Kingma and Jimmy Lei Ba. 2015. Adam: A method for stochastic optimization.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 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.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *ACL (1)*.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *EMNLP (Demonstration)*.

Surafel Melaku Lakew, Mauro Cettolo, and Marcello Federico. 2018. A comparison of transformer and recurrent neural networks on multilingual neural machine translation. *international conference on computational linguistics*.

Jindřich Libovický and Alexander Fraser. 2021. Findings of the WMT 2021 shared tasks in unsupervised MT and very low resource supervised MT. In *Proceedings of the Sixth Conference on Machine Translation*, pages 726–732, Online. Association for Computational Linguistics.

Myle Ott, Michael Auli, David Grangier, and Marc'Aurelio Ranzato. 2018. Analyzing uncertainty in neural machine translation. In *ICML*.

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.

Surangika Ranathunga, En-Shiun Annie Lee, Marjana Prifti Skenduli, Ravi Shekhar, Mehreen Alam, and Rishemjit Kaur. 2021. Neural machine translation for low-resource languages: A survey. *arXiv: Computation and Language*.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826. IEEE.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Daimeng Wei, Zongyao Li, Zhanglin Wu, Zhengzhe Yu, Xiaoyu Chen, Hengchao Shang, Jiaxin Guo, Minghan Wang, Lizhi Lei, Min Zhang, et al. 2021. Hw-tsc's participation in the wmt 2021 news translation shared task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 225–231.

Lijun Wu, Juntao Li, Yue Wang, Qi Meng, Tao Qin, Wei Chen, Min Zhang, Tie-Yan Liu, et al. 2021. R-drop: Regularized dropout for neural networks. *Advances in Neural Information Processing Systems*, 34:10890–10905.

Lijun Wu, Yiren Wang, Yingce Xia, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. 2019. Exploiting monolingual data at scale for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4207–4216.