Language Branch Gated Multilingual Neural Machine Translation

Haoran Sun and Deyi Xiong[*]
College of Intelligence and Computing, Tianjin University, Tianjin, China
hrsun0105@gmail.com, dyxiong@tju.edu.cn

Abstract

Knowledge transfer across languages is crucial for multilingual neural machine translation. In this paper, we propose language branch (LB) gated multilingual neural machine translation that encourages knowledge transfer within the same language branch with a LB-gated module that is integrated into both the encoder and decoder. The LB-gated module distinguishes LB-specific parameters from global parameters shared by all languages and routes languages from the same LB to the corresponding LB-specific network. Comprehensive experiments on the OPUS-100 dataset show that the proposed approach substantially improves translation quality on both middle- and low-resource languages over previous methods. Further analysis demonstrates its ability in learning similarities between language branches.

1 Introduction

Recent years have witnessed a growing interest in multilingual neural machine translation (NMT), which supports translation among multiple languages with one single model (Dong et al., 2015; Luong et al., 2016; Firat et al., 2016; Johnson et al., 2017; Aharoni et al., 2019; Arivazhagan et al., 2019; Xu et al., 2021). As the parameters of multilingual NMT are fully or partially shared by multiple languages, knowledge transfer across languages improves translation quality of low-resource languages.

Despite these advantages, there still exist challenges in multilingual NMT. As previous studies have found, multilingual NMT usually underperforms its bilingual counterparts on high-resource languages (Johnson et al., 2017; Arivazhagan et al., 2019). A way to alleviate this issue is to use language-aware modules, which could be more computationally efficient than simply enlarging model capacity with deeper/wider models (Zhang et al., 2020, 2021; Lin et al., 2021). As for low-resource languages, positive transfer is more pronounced among related languages than distant languages (Sachan and Neubig, 2018). Additionally, explicit language clustering benefits multilingual NMT (Tan et al., 2019) although itself can learn linguistic typology during training (Lu et al., 2018).

Inspired by these, we propose language branch (subfamily) gated multilingual neural machine translation, which fuses language branch information into multilingual NMT by a **L**anguage **B**ranch **G**ated **M**odule (LBGM). A language branch is a subfamily of a language family. Take the Indo-European language family as an example. It can be further divided into subfamilies like Germanic, Slavic, Celtic, etc. The reason why we use language branches rather than language families is that the latter are relatively coarse-grained. Languages within a language branch are more closely related to each other than those in a language family.

A token that indicates the language branch for the current sentence is fed into LBGM, in addition to the input from other layers in multilingual NMT. With the language branch token, LBGM distinguishes language-branch-specific parameters from global parameters shared by all languages and uses a gate to aggregate these two parts as the output of the module.

We conduct experiments on the OPUS-100 dataset (Zhang et al., 2020) with a large number of different languages. Our main findings can be summarized as follows:

- The proposed LBGM can significantly improve translation quality, achieving more substantial gains for middle- and low-resource languages than its counterparts.
- LBGM performs better for language branches that contain plentiful languages, including

[*]Corresponding author.

Proceedings of the 29th International Conference on Computational Linguistics, pages 5046–5053
October 12–17, 2022.

not only high-resource languages, but also middle/low-resource languages.

- LBGM is capable of capturing similarities between language branches.

2 Related Work

Research of multilingual NMT mainly focuses on partial or full parameter sharing in NMT modules (Dong et al., 2015; Luong et al., 2016; Firat et al., 2016). Johnson et al. (2017) propose prefixing sentences with a language token in a joint set of parallel corpora, using a single NMT model to enable multilingual translation. There is a trade-off between boosting the performance of low-resource languages and sacrificing the performance of high-resource languages (Arivazhagan et al., 2019).

Language-Specific Parameters Previous works have been trying to use language-specific parameters to alleviate the trade-off issue in multilingual NMT, such as adding adaptation layers into pre-trained models for each language (Bapna and Firat, 2019; Philip et al., 2020; Zhu et al., 2021). Zhang et al. (2020) propose language-aware layer normalization to relax normalization constraint for target languages. Lin et al. (2021) produce masks for different language pairs and use them to select subnetworks for language pairs, in order to counter parameter interference. The closest work to ours is done by Zhang et al. (2021), who introduce a module called CLSR into the Transformer model. The CLSR module adopts a gating function, which is trained with injected zero-mean Gaussian noise and discretized at inference time, to choose whether to share the parameters for all languages or not. However, the CLSR does not take the relationship between languages into account, since it uses language identity as the router.

Language Clustering Sachan and Neubig (2018) have found that full parameter sharing improves translation quality mainly for related languages that are from the same language group. Tan et al. (2019) attempt to cluster languages into different groups using two methods: prior knowledge and language embedding. They build a multilingual NMT model for each group and observe that both clustering approaches are able to improve model performance. Fan et al. (2021) propose adding a language-specific layer for each language group. They cluster languages according to the amount of training data and vocabulary. Different from them, we use a linguistically-motivated and fine-grained method to group languages (i.e., language subfamily).

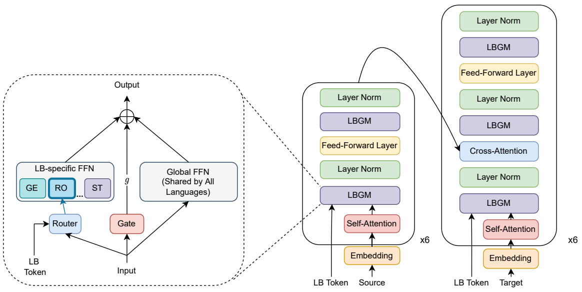

Figure 1: Illustration of the proposed LBGM model.

3 Methodology

We adopt the Transformer model as the backbone network (Vaswani et al., 2017). Following Johnson et al. (2017), we extend the Transformer to multilingual Transformer by prefixing a language token to the source and target side respectively.

We cluster languages into their language branches. More specifically, for Indo-European languages, we cluster them into 10 branches, including Baltic, Celtic, East Slavic, Germanic, Hellenic, Indo-Aryan, Iranian, Romance, South Slavic and West Slavic. But for Afro-Asiatic languages, as there are only five languages in the dataset we use, we just group them into one language branch. For isolated languages, e.g., Esperanto, Japanese, we keep them in the isolated language group, expecting them to benefit from the positive transfer from all languages.

We adapt the multilingual Transformer to integrate language branch information with the proposed LBGM, which is illustrated in Figure 1. Particularly, we use an additional token, i.e., LB token, to indicate the language branch for each corresponding sentence. LBGM contains a gating function and two feedforward networks, namely, LB-specific FFN and global FFN. The LB-specific FFN is exclusively used for sentences from the corresponding language branch while the global FFN is fully shared across all languages. The purpose of the fully-shared global FFN is to capture global linguistic information from all language pairs, so as to enable knowledge transfer across all languages, especially for the isolated language branch which only contains one or a few languages. The LB-specific FFN is to capture linguistic features for languages from the same language branch and hence enabling intra-language-branch knowledge transfer. LBGM can be formulated as follows:

$$\mathrm{Gate}(\boldsymbol{a}^l) = \mathrm{ReLU}(\boldsymbol{a}^l \boldsymbol{W} + b) \tag{1}$$

$$g^l = \sigma(\mathrm{Gate}(\boldsymbol{a}^l)) \tag{2}$$

$$\begin{aligned}\mathrm{LBGM}(\boldsymbol{a}^l) &= g^l \cdot \mathrm{FFN}^{\text{LB-specific}}(\boldsymbol{a}^l) \\ &+ (1-g^l) \cdot \mathrm{FFN}^{\text{global}}(\boldsymbol{a}^l)\end{aligned} \tag{3}$$

where $\boldsymbol{W}$ and b are trainable parameters, $\boldsymbol{a}^l$ is the output from the preceding layer l. FFNs are calculated as follows:

$$\mathrm{FFN}^{\text{LB-specific}} = \boldsymbol{a}^l \boldsymbol{W}^{\text{LB-specific}} + \boldsymbol{b}^{\text{LB-specific}} \tag{4}$$

$$\mathrm{FFN}^{\text{global}} = \boldsymbol{a}^l \boldsymbol{W}^{\text{global}} + \boldsymbol{b}^{\text{global}} \tag{5}$$

The LB token is used to route information from previous layer into the corresponding LB-specific FFN, which is acting as an “expert” for that language branch.

In contrast to the previous CLSR method Zhang et al. (2021), we do not use the discrete gating function and additional loss component. In our preliminary experiments, we have found that these two components could not bring improvements to our LBGM module and the discretized gating function is even harmful to the LBGM. Instead, we use an individual gating function per LBGM sub-module (not shared by the whole model like CLSR). This is because, at different positions, the importance of the two types of FFN may be different, which is to be determined by the corresponding gating function in our model.

4 Experiments

We conducted experiments with a massive number of languages to examine the effectiveness of the proposed LBGM.

4.1 Settings

We used the OPUS-100 dataset (Zhang et al., 2020) for our experiments. OPUS-100[1] is an English-centric dataset covering 99 language pairs. As 5 language pairs do not have their test and dev sets, we conducted experiments using the rest 94 language pairs. We roughly divided the languages into three categories according to the training data size: high-resource languages (more than 1M training samples, 44 languages), low-resource languages (fewer than 0.1M training samples, 21 languages) and middle-resource languages (others, 29 languages) following Zhang et al. (2020). This division is only for experiments while in our model, we linguistically grouped these languages into 26 language branches, shown in Table 5 in Appendix A.

We applied Byte Pair Encoding (BPE) (Sennrich et al., 2016) to preprocess the data with a joint vocabulary size of 64K, using the SentencePiece Toolkit (Kudo and Richardson, 2018)[2]. We adopted the temperature-based oversampling method with a temperature of $T = 5$.

Translation quality was evaluated by BLEU (Papineni et al., 2002) using SacreBleu (Post, 2018)[3].

We adopted the Transformer-base model (Vaswani et al., 2017) as our baseline. The dimensions of our LB-specific FFNs and global FFN are 512. As we have 26 language branches, so the total number of LB-specific FFNs is 26. We also compared with CLSR (Zhang et al., 2021), which uses a language-specific module and conditional routing function to learn the representation of each language. All models were implemented with fairseq (Ott et al., 2019)[4].

Other details about experiments and model settings are in Appendix A.

4.2 Main Results of One-to-Many and Many-to-One Translation

We first conducted experiments for one-to-many and many-to-one translation (i.e., English→X and X→English). Although there is only one language on the source side for one-to-many translation and on the target side for many-to-one translation, we use the proposed LBGM in both the encoder and decoder, as shown in Figure 1, to keep both LB-informed.

As shown in Table 1, our LBGM outperforms the baseline and CLSR on the OPUS-100 dataset. Particularly, we achieve an overall improvement of 1.32/0.70 BLEU points on English→X, 1.31/0.30 BLEU points on X→English over the baseline/CLSR. In terms of the amount of training data available, we observe that the proposed LBGM gains larger improvements on low-resource languages than those on high/middle-resource languages over the baseline and CLSR.

[1] http://opus.nlpl.eu/opus-100.php

[2] https://github.com/google/sentencepiece

[3] https://github.com/mjpost/sacrebleu

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

Model	#Params	English→X			
		High	Mid	Low	All
Baseline	99M	15.01	22.26	25.99	19.70
Baseline-592dim	117M	15.44	22.72	26.41	20.14
CLSR	154M	15.81	22.73	26.43	20.32
LBGM-LS	154M	**16.01**	23.13	26.87	20.63
LBGM	117M	15.95	**23.77**	**27.85**	**21.02**

Model	#Params	X→English			
		High	Mid	Low	All
Baseline	99M	20.71	23.15	25.33	22.49
Baseline-592dim	117M	21.68	23.69	25.80	23.22
CLSR	154M	21.82	24.22	26.07	23.50
LBGM-LS	154M	**22.09**	23.77	26.10	23.51
LBGM	117M	22.04	**24.58**	**26.42**	**23.80**

Table 1: Results on the OPUS-100 dataset. We report the average BLEU of English→X and X→English translation on 94 language pairs.

4.3 Ablation Study

In order to eliminate the difference in the number of parameters of the LBGM and baseline, we scaled the hidden size of the baseline from 512 to 592 dimensions, denoted in Table 1 as Baseline-592dim. The results still demonstrate that our LBGM is more efficient than the vanilla Transformer. This ablation study confirms that the improvement obtained by the LBGM is not due to the capacity advantage.

We conducted another group of experiments to investigate whether language branches are helpful in comparison to the original CLSR architecture, which could tell us whether the modified LBGM architecture would be better adapted to the use of language branches. We used the same settings as the previous experiments. Differently, language identities, instead of language branches, were used as the router. The results are shown in Table 1 denoted as LBGM-LS. From the comparison between the LBGM-LS and LBGM, we can find that language branches are indeed important. Although the LBGM-LS obtains a slightly higher improvement than the LBGM on high-resource languages (0.06 BLEU), it's acceptable for our LBGM approach. That is because the LBGM-LS allocates an individual FFN module for each language, rather than one for each language branch, and it alleviates the capacity constraints of the model on high-resource languages. LBGM-LS still outperforms the CLSR approach, suggesting that our modifications to the original CLSR module are not harmful to the model but more appropriate in the context of using language branches.

Lang	Data Size	Baseline	CLSR	LBGM	Δ-B	Δ-C
da	1000000	20.74	22.45	**22.64**	1.90	0.19
de	1000000	16.90	17.79	**18.03**	1.13	0.24
is	1000000	10.83	11.86	**12.01**	1.18	0.15
nl	1000000	16.58	17.34	**17.86**	1.28	0.52
no	1000000	17.81	18.92	**19.34**	1.53	0.42
sv	1000000	18.37	19.27	**19.82**	1.45	0.55
nn	486055	23.55	24.88	**25.40**	1.85	0.52
af	275512	30.02	31.14	**32.15**	2.13	1.01
nb	142906	23.16	24.39	**25.48**	2.32	1.09
fy	54342	25.03	27.02	**27.63**	2.60	0.61
li	25535	27.42	29.27	**30.59**	3.17	1.32
yi	15010	25.29	27.31	**30.73**	5.44	3.42

Table 2: Results on the English→X translation for the Germanic language branch, which includes high/middle/low-resource language pairs. Δ-B and Δ-C denote the improvements over Baseline and CLSR respectively.

LB	Lang	Data Size	Bilingual	Baseline	CLSR	LBGM
WS	cs	1000000	**18.03**	14.92	15.46	16.11
	pl	1000000	**15.37**	11.57	11.83	12.35
	sk	1000000	**18.66**	15.89	16.71	17.44
BA	lt	1000000	17.63	18.55	**19.82**	19.61
	lv	1000000	21.10	20.83	**22.57**	22.09

Table 3: Results on the West Slavic (WS) language branch and Baltic (BA) language branch on English→X translation. Bilingual is the bilingual model trained with the same architecture.

4.4 Effect on the Different Types of Language Branches

To further investigate how the proposed LBGM improves translation quality, we categorize language branches into three types: (I) language branch containing high/middle/low-resource languages. (II) language branch with only high-resource languages. (III) language branch with one or two languages, which are usually isolated languages. We analyzed the effects of our LBGM on these three types of language branches.

Table 2 shows the results of the Germanic language branch, a Type-I language branch as mentioned above. We list the languages (denoted by their ISO-639-1 codes) in descending order of the amount of training data. On this language branch type, LBGM outperforms both the baseline and CLSR on all languages with different levels of resource. Particularly, as the amount of training data decreases, the improvements over the baseline and CLSR increase.

For the Type-II language branch (i.e., including only high-resource languages), we show the results of both West Slavic and Baltic language branches on English→X translation in Table 3. As these language branches include only high-resource lan-

Lang	Data Size	Baseline	CLSR	LBGM
ja	1000000	4.73	**5.27**	4.46
ko	1000000	**3.03**	2.91	2.84
ka	377306	14.39	**14.91**	14.57

Table 4: Results on the isolated language branch of English→X translation.

guages, negative transfer usually happens (compared to bilingual models). Fortunately, on this type of language branches, our LBGM still substantially outperforms the baseline and is better than or comparable with CLSR.

Finally, Table 4 shows results on the isolated language branch. It can be seen that LBGM achieves results comparable to CLSR. And the effect on isolated languages depends more on the characteristics of the language itself.

All these results suggest that our LBGM can significantly improve performance on both middle- and low-resource languages and achieve comparable results to CLSR on high-resource languages but with fewer parameters. The gains on middle- and low-resource languages are more substantial when the language branch contains mixed languages in terms of the amount of available training data.

5 Analysis

Ideally, the closer two language branches are to each other, the more similar the parameters of the corresponding LB-specific FFNs are to each other in LBGM, especially for language branches which are from the same language family. Motivated by this, we calculated the pairwise cosine similarities of LB-specific FFNs of any two different language branches learned in LBGM, trying to use these similarities to measure the relationship between language branches. Specifically, the cosine similarity is computed with the weight matrix of LB-specific FFN (reshaped into a vector via row-wise concatenation) as follows:

$$\text{Cosine Similar}(\boldsymbol{V_1}, \boldsymbol{V_2}) = \frac{\boldsymbol{V_1} \cdot \boldsymbol{V_2}}{\|\boldsymbol{V_1}\| \|\boldsymbol{V_2}\|} \qquad (6)$$

where $\| \cdot \|$ denotes the L_2 norm, $\boldsymbol{V_1}$ and $\boldsymbol{V_2}$ are vectors reshaped from weight matrices.

Figure 2 shows the cosine similarity matrix of different language branches (26 in total contained in the OPUS-100 dataset) learned by our LBGM. The deeper the color is, the more similar the two language branches are. It is clear to see that GE (Germanic) is more similar to BA (Baltic), WS (West Slavic), CE (Celtic), etc., which are from the same language family, than other language branches, indicated by deeper color in Figure 2. Similar results can be observed among JA (Japonic), KO (Koreanic) and ST (Sino-Tibetan) language branches, among OG (Oghuz), KAL (Karluk) and KI (Kipchak) language branches, etc. This suggests that the proposed LBGM is able to learn similarities between different language branches.

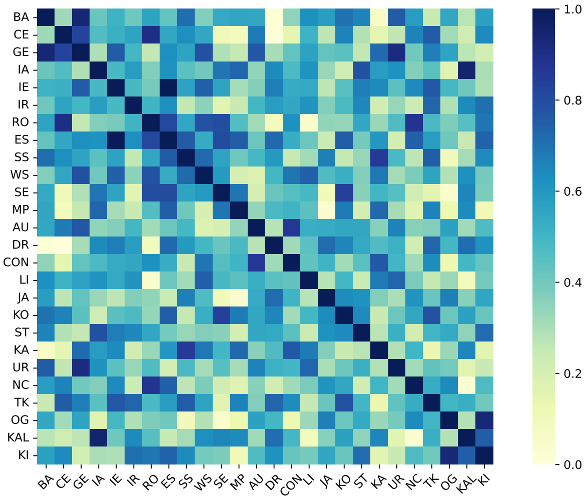

Figure 2: Pairwise language branch similarities learned by LBGM with one-to-many translation. Better view with color.

6 Conclusions

In this paper, we have presented LBGM that uses a LB-specific FFN and a global FFN shared across all languages to enhance knowledge transfer within the same language branch for multilingual neural machine translation. Experiments on the OPUS-100 dataset have shown that LBGM can significantly improve translation quality on both middle- and low-resource languages, over the baseline and CLSR (Zhang et al., 2021). Further analysis on LB-specific FFN discloses that the proposed LBGM is able to capture language branch relations.

Acknowledgments

The present research was supported by the Key Research and Development Program of Yunnan Province (Grant No. 202203AA080004-2). We would like to thank the anonymous reviewers for their insightful comments.

References

Roee Aharoni, Melvin Johnson, and Orhan Firat. 2019. Massively multilingual neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3874–3884, Minneapolis, Minnesota. Association for Computational Linguistics.

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George F. Foster, Colin Cherry, Wolfgang Macherey, Zhifeng Chen, and Yonghui Wu. 2019. Massively multilingual neural machine translation in the wild: Findings and challenges. *CoRR*, abs/1907.05019.

Ankur Bapna and Orhan Firat. 2019. Simple, scalable adaptation 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 1538–1548, Hong Kong, China. Association for Computational Linguistics.

Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. 2015. Multi-task learning for multiple language translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1723–1732, Beijing, China. Association for Computational Linguistics.

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, Michael Auli, and Armand Joulin. 2021. Beyond english-centric multilingual machine translation. *Journal of Machine Learning Research*, 22(107):1–48.

Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. 2016. Multi-way, multilingual neural machine translation with a shared attention mechanism. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 866–875, San Diego, California. Association for Computational Linguistics.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

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: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Zehui Lin, Liwei Wu, Mingxuan Wang, and Lei Li. 2021. Learning language specific sub-network for multilingual machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 293–305, Online. Association for Computational Linguistics.

Yichao Lu, Phillip Keung, Faisal Ladhak, Vikas Bhardwaj, Shaonan Zhang, and Jason Sun. 2018. A neural interlingua for multilingual machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 84–92, Brussels, Belgium. Association for Computational Linguistics.

Minh-Thang Luong, Quoc V. Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2016. Multi-task sequence to sequence learning. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

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, Minneapolis, Minnesota. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Jerin Philip, Alexandre Berard, Matthias Gallé, and Laurent Besacier. 2020. Monolingual adapters for zero-shot neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4465–4470, Online. Association for Computational Linguistics.

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, Brussels, Belgium. Association for Computational Linguistics.

Devendra Sachan and Graham Neubig. 2018. Parameter sharing methods for multilingual self-attentional translation models. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 261–271, Brussels, Belgium. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Xu Tan, Jiale Chen, Di He, Yingce Xia, Tao Qin, and Tie-Yan Liu. 2019. Multilingual neural machine translation with language clustering. 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 963–973, Hong Kong, China. Association for Computational Linguistics.

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 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 5998–6008.

Hongfei Xu, Qiuhui Liu, Josef van Genabith, and Deyi Xiong. 2021. Modeling task-aware MIMO cardinality for efficient multilingual neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 361–367, Online. Association for Computational Linguistics.

Biao Zhang, Ankur Bapna, Rico Sennrich, and Orhan Firat. 2021. Share or not? learning to schedule language-specific capacity for multilingual translation. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*.

Biao Zhang, Philip Williams, Ivan Titov, and Rico Sennrich. 2020. Improving massively multilingual neural machine translation and zero-shot translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1628–1639, Online. Association for Computational Linguistics.

Yaoming Zhu, Jiangtao Feng, Chengqi Zhao, Mingxuan Wang, and Lei Li. 2021. Counter-interference adapter for multilingual machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2812–2823, Punta Cana, Dominican Republic. Association for Computational Linguistics.

ISO	Name	Family	Branch	Code
am	Amharic	Afro-Asiatic	Semitic	SE
ar	Arabic	Afro-Asiatic	Semitic	SE
ha	Hausa	Afro-Asiatic	Chadic	SE
he	Hebrew	Afro-Asiatic	Semitic	SE
mt	Maltese	Afro-Asiatic	Semitic	SE
km	Khmer	Austroasiatic	Austroasiatic	AU
vi	Vietnamese	Austroasiatic	Austroasiatic	AU
id	Indonesian	Austronesian	Malayo-Polynesian	MP
mg	Malagasy	Austronesian	Malayo-Polynesian	MP
ms	Malay	Austronesian	Malayo-Polynesian	MP
eo	Esperanto	Constructed	Constructed	CON
kn	Kannada	Dravidian	Dravidian	DR
ml	Malayalam	Dravidian	Dravidian	DR
ta	Tamil	Dravidian	Dravidian	DR
te	Tegulu	Dravidian	Dravidian	DR
lt	Lithuanian	Indo-European	Baltic	BA
lv	Latvian	Indo-European	Baltic	BA
br	Breton	Indo-European	Celtic	CE
cy	Welch	Indo-European	Celtic	CE
ga	Irish	Indo-European	Celtic	CE
gd	Scots Gaelic	Indo-European	Celtic	CE
be	Byelorussian	Indo-European	East Slavic	ES
ru	Russian	Indo-European	East Slavic	ES
uk	Ukrainian	Indo-European	East Slavic	ES
af	Afrikaans	Indo-European	Germanic	GE
da	Danish	Indo-European	Germanic	GE
de	German	Indo-European	Germanic	GE
fy	Frisian	Indo-European	Germanic	GE
is	Icelandic	Indo-European	Germanic	GE
li	Limburgan	Indo-European	Germanic	GE
nb	Bokmål	Indo-European	Germanic	GE
nl	Dutch	Indo-European	Germanic	GE
nn	Nynorsk	Indo-European	Germanic	GE
no	Norwegian	Indo-European	Germanic	GE
sv	Swedish	Indo-European	Germanic	GE
yi	Yiddish	Indo-European	Germanic	GE
as	Assamese	Indo-European	Indo-Aryan	IA
bn	Bengali	Indo-European	Indo-Aryan	IA
gu	Gujarati	Indo-European	Indo-Aryan	IA
hi	Hindi	Indo-European	Indo-Aryan	IA
mr	Marathi	Indo-European	Indo-Aryan	IA
ne	Nepali	Indo-European	Indo-Aryan	IA
or	Oriya	Indo-European	Indo-Aryan	IA
pa	Punjabi	Indo-European	Indo-Aryan	IA
si	Singhalese	Indo-European	Indo-Aryan	IA
ur	Urdu	Indo-European	Indo-Aryan	IA
el	Greek	Indo-European	Hellenic	HE
sq	Albanian	Indo-European	Albanian	HE
fa	Persian	Indo-European	Iranian	IR
ku	Kurdish	Indo-European	Iranian	IR
ps	Pashto	Indo-European	Iranian	IR
tg	Tajik	Indo-European	Iranian	IR
ca	Catalan	Indo-European	Romance	RO
es	Spanish	Indo-European	Romance	RO
fr	French	Indo-European	Romance	RO
gl	Galician	Indo-European	Romance	RO
it	Italian	Indo-European	Romance	RO
oc	Occitan	Indo-European	Romance	RO
pt	Portuguese	Indo-European	Romance	RO
ro	Romanian	Indo-European	Romance	RO
wa	Walloon	Indo-European	Romance	RO
bg	Bulgarian	Indo-European	South Slavic	SS
bs	bosanski	Indo-European	South Slavic	SS
hr	Croatian	Indo-European	South Slavic	SS
mk	Macedonian	Indo-European	South Slavic	SS
sh	Serbo-Croatian	Indo-European	South Slavic	SS
sl	Slovenian	Indo-European	South Slavic	SS
sr	Serbian	Indo-European	South Slavic	SS
cs	Czech	Indo-European	West Slavic	WS
pl	Polish	Indo-European	West Slavic	WS
sk	Slovak	Indo-European	West Slavic	WS
ja	Japanese	Japonic	Japonic	JA
ka	Georgian	Kartvelian	Kartvelian	KA
ko	Korean	Koreanic	Koreanic	KO
eu	Basque	Language isolate	Language isolate	LI
ig	Igbo	Niger–Congo	Niger–Congo	NC
rw	Kinyarwanda	Niger–Congo	Niger–Congo	NC
xh	Xhosa	Niger–Congo	Niger–Congo	NC
zu	Zulu	Niger–Congo	Niger–Congo	NC
my	Burmese	Sino-Tibetan	Sino-Tibetan	ST
zh	Chinese	Sino-Tibetan	Sino-Tibetan	ST
th	Thai	Tai–Kadai	Tai–Kadai	TK
ug	Uigur	Turkic	Karluk	KAL
uz	Uzbek	Turkic	Karluk	KAL
kk	Kazakh	Turkic	Kipchak	KI
ky	Kirghiz	Turkic	Kipchak	KI
tt	Tatar	Turkic	Kipchak	KI
az	Azerbaijani	Turkic	Oghuz	OG
tk	Turkmen	Turkic	Oghuz	OG
tr	Turkish	Turkic	Oghuz	OG
et	Estonian	Uralic	Uralic	UR
fi	Finnish	Uralic	Uralic	UR
hu	Hungarian	Uralic	Uralic	UR
se	Northern Sami	Uralic	Uralic	UR

Table 5: ISO-639-1 language code, language name, language family, language branch and language branch code in the OPUS-100 dataset.

A Appendix

A.1 Languages in the OPUS-100 Dataset

We list the languages used in our experiments in Table 5. The language branches are most based on linguistic characteristics, but some of them are based on geopolitical locations, e.g., Greek and Albanian.

A.2 Experiment Settings

The Transformer-base model has 6 layers for both encoder and decode, 8 attention heads and 512 dimensions for embeddings, 2048 dimensions for FFN layer. We set the dropout rate to 0.1 for all modules. The hyperparameters of our LBGM were the same as the Transformer-base model. The gate function is implemented by FFN which input dimension is 512 and output dimension is 1.

We optimized parameters using Adam optimizer (Kingma and Ba, 2015) with a label smoothing rate of 0.1. The learning rate was scheduled according to the inverse square root of running steps with a warmup step of 4K and the weight decay rate was set to 0.0001. We set the maximum number of steps to 500K. For inference, we used beam search with a beam size of 4 and a length penalty of 0.6.