Integrating Multi-scale Contextualized Information for Byte-based Neural Machine Translation

Langlin Huang [1,3], **Yang Feng** [1,2,3*]
[1] Key Laboratory of Intelligent Information Processing
Institute of Computing Technology, Chinese Academy of Sciences
[2] Key Laboratory of AI Safety, Chinese Academy of Sciences
[3] University of Chinese Academy of Sciences
{huanglanglin21s, fengyang}@ict.ac.cn

Abstract

Subword tokenization is a common method for vocabulary building in Neural Machine Translation (NMT) models. However, increasingly complex tasks have revealed its disadvantages. First, a vocabulary cannot be modified once it is learned, making it hard to adapt to new words. Second, in multilingual translation, the imbalance in data volumes across different languages spreads to the vocabulary, exacerbating translations involving low-resource languages. While byte-based tokenization addresses these issues, byte-based models struggle with the low information density inherent in UTF-8 byte sequences. Previous works enhance token semantics through local contextualization but fail to select an appropriate contextualizing scope based on the input. Consequently, we propose the Multi-Scale Contextualization (MSC) method, which learns contextualized information of varying scales across different hidden state dimensions. It then leverages the attention module to dynamically integrate the multi-scale contextualized information. Experiments show that MSC significantly outperforms subword-based and other byte-based methods in both multilingual and out-of-domain scenarios. We have uploaded the code to github[1].

1 Introduction

In neural machine translation (NMT) systems, subword tokenization has been the most common and effective method to mitigate the out-of-vocabulary (OOV) problem. However, both BPE (Sennrich et al., 2016) and SentencePiece (Kudo and Richardson, 2018) fix the word segmentation rule or vocabulary once they have learned them on the initial corpus, making it difficult to ensure adaptation to new corpora. This is worsened in out-of-domain scenarios. Additionally, in multilingual scenarios with data imbalance, subword vocabularies tend to focus on high-resource languages, overlooking low-resource ones. This imbalance can cause an increase in OOV cases or over-segmentation of texts, harmful to translation model performance.

Byte-based method is able to solve these problems with few embedding parameters and has aroused extensive researches (Wang et al., 2020; Shaham and Levy, 2021; Xue et al., 2022; Yu et al., 2023; Edman et al., 2023; Sreedhar et al., 2023). In byte-based models, text is converted into byte sequences according to UTF-8 encoding, with each byte as a token within the vocabulary. They generally use a vocabulary with a maximum size of 256 but can adapt to imbalanced scenarios like multilingual translation and out-of-domain adaptation.

However, a feature of UTF-8 encoding hinders conventional Transformer model (Vaswani et al., 2017) from adapting well to byte-based vocabulary: a single character may correspond to 1 to 4 UTF-8 bytes. The number is 1 for English characters, but Arabic and many Asian languages require multiple bytes to represent a single character. Therefore, sometimes a single byte does not have a determined meaning; it requires the integration of local information to encode its semantics. To address that, various methods have been proposed for integrating local contextual information. SU4MT (Huang et al., 2023) learns contextual information with an Attentive Semantic Fusion layer, but requires accurate segmentation. MEGABYTE (Yu et al., 2023) segments a sentence into blocks of 4 and simply concatenates the tokens. Charformer (Tay et al., 2022) segments a sentence 4 times with block sizes ranging from 1 to 4 each, and employs mean-pooling to perform local integration. The weighted-sum of 4 results yields the final output. LOBEF (Sreedhar et al., 2023) proposed Byte-*n*CF, replacing mean-pooling with Convolutional Neural Networks (CNNs) for better performance.

Though these methods learn and leverage con-

*Corresponding author.
[1] https://github.com/ictnlp/Multiscale-Contextualization

Findings of the Association for Computational Linguistics: ACL 2024, pages 9794–9801

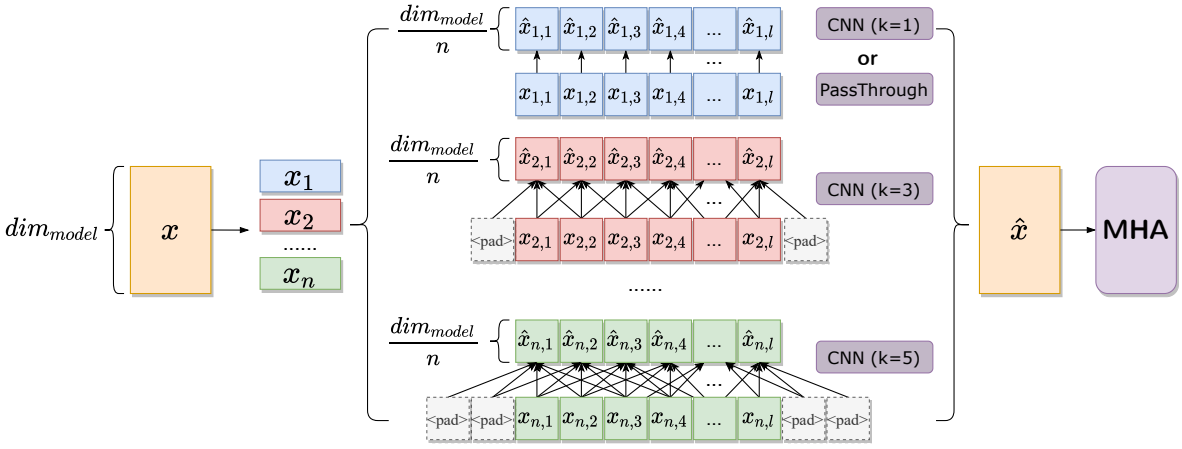

Figure 1: **Multi-Scale Contextualization** module: the input vector x, with hidden state dimension dim_{model} and text length l, is divided into n parts according to the hidden state dimensions, and then n contextualizing functions with different scopes process these parts respectively. The output $\hat{x}$ now contains multi-scale information and acts as input to the Multi-Head Attention module.

textual information in larger scales, they are limited by fixed block sizes and can not adjust the fusion weights according to the scripts of different languages. To remedy this, we propose the Multi-Scale Contextualization (MSC) method, which firstly learn contextual information of multiple scales in different hidden state dimensions, and secondly leverage the attention mechanism to fuse multi-scale information with dynamic weights. Our method better adapts to complex input scripts by allowing the model to adaptively fuse information of different granularities based on the varying content of the inputs. Experimental results demonstrate that our MSC method exhibits superior adaptability across various languages and datasets.

2 Method

In this section, we introduce the proposed Multi-Scale Contextualization (MSC) method. Byte-based models usually learn contextualized information implicitly. What MSC does is explicitly modeling the contextual information of multiple scales by grouping the hidden state dimensions and let different parts of the dimensions learn information of different scales.

Specifically, we insert a multi-scale contextualization module right before the Multi-Head Attention (MHA) module, as is depicted in Fig 1. The input vector $\mathbf{x}$ is divided according to the hidden state dimension into n parts $[\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n]$.

Then, n contextualizing functions $g(\cdot)$ are applied to these parts respectively. A simple and effective structure for local contextualization is the 1-D convolution neural network (CNN). Therefore, we leverage CNNs with different kernel size k to control the contextualization scope. Since different dimensions are contextualized with various granularities, our model can realize multi-scale contextualization. To preserve the original information, $g(\cdot)$ is also allowed to be a "PassThrough" function, which directly returns the input without any operations.

$$g_i(\cdot, k) = \begin{cases} \text{PassThrough}(\cdot) & , k = 0 \\ \text{CNN}(\cdot, k) & , k > 0 \end{cases} \quad (1)$$

In equation (1), $g_i(\cdot, k)$ means the contextualization function for group i, and k is the kernel size. Here, k=0 denotes the "PassThrough" function for simplification.

Next, the contextualized vector parts $\hat{x}_i$ are calculated by $g_i(x_i, k)$. Finally, they are concatenated to form $\hat{x}$, which acts as the input of the MHA module.

Preliminary experiments on CNN configurations guided the best structure for the whole model. First, padding on the left side deteriorates model performance, so the CNNs are set to pad on both sides, as shown in Figure 1. Second, applying MSC to Transformer decoder layers causes a discrepancy between training and testing, when a token's right side tokens are not yet generated. As a result, MSC is only applied to encoder layers.

It is worth noting that the kernel size k is recommended to be an odd number or zero, otherwise, manual zero-padding is required to keep the output length the same as the input. Empirically, it is

LID	Subword			Byte		
	Learned (60.6M)	mBART (172.2M)	Aharoni* (∼93M)	Transformer (44.3M)	Byte-*n*CF (47.0M)	MSC (45.0M)
Az	11.58	10.24	11.24	$12.61_{(\pm 0.42)}$	$12.46_{(\pm 0.27)}$	**13.24**$_{(\pm 0.25)}$
Be	18.83	15.41	18.28	$21.41_{(\pm 0.15)}$	$21.51_{(\pm 0.19)}$	**22.10**$_{(\pm 0.49)}$
Gl	26.81	28.20	28.63	$31.08_{(\pm 0.43)}$	$31.44_{(\pm 0.05)}$	**31.98**$_{(\pm 0.13)}$
Sk	24.93	24.76	26.78	$27.64_{(\pm 0.26)}$	$27.65_{(\pm 0.35)}$	**28.39**$_{(\pm 0.27)}$
AVG-LR	20.54	19.65	21.23	$23.19_{(\pm 0.04)}$	$23.26_{(\pm 0.14)}$	**23.93**$_{(\pm 0.14)}$
Ar	23.35	22.57	25.93	$25.60_{(\pm 0.13)}$	$25.89_{(\pm 0.29)}$	**26.33**$_{(\pm 0.09)}$
De	26.33	27.78	28.87	$30.14_{(\pm 0.49)}$	$30.56_{(\pm 0.33)}$	**31.02**$_{(\pm 0.21)}$
He	27.09	26.59	30.19	$30.38_{(\pm 0.33)}$	$30.55_{(\pm 0.17)}$	**31.29**$_{(\pm 0.19)}$
It	28.45	30.36	32.42	$32.97_{(\pm 0.66)}$	$33.34_{(\pm 0.28)}$	**33.63**$_{(\pm 0.34)}$
AVG-HR	26.31	26.83	29.35	$29.77_{(\pm 0.40)}$	$30.08_{(\pm 0.26)}$	**30.57**$_{(\pm 0.13)}$
AVG-58	21.44	21.15	-	$23.63_{(\pm 0.16)}$	$23.70_{(\pm 0.21)}$	**24.30**$_{(\pm 0.10)}$

Table 1: The experiment results on TED-59 dataset, measured by SacreBLEU. The table includes 4 low-resource (LR) and 4 high-resource (HR) languages selected by Aharoni et al. (2019). The "*" sign denotes the results are cited from Aharoni et al. (2019). Byte-based models are experimented three times and the table shows the average scores and the standard deviation.

better to choose k from {0, 1, 3, 5, 7}.

3 Experiments

We experiment with two multilingual datasets and a domain-adaptation dataset to investigate the performances and properties of the MSC approach and other byte-based language models.

3.1 Datasets

Multilingual Many-to-One Translation

We use a multilingual TED corpus of 59 languages (Qi et al., 2018), TED-59, which includes both high and low-resource languages. All cases are English-centered. We collect the raw data from Salesky et al. (2023) and preprocess it with two subword-level vocabularies and a byte-level one. For subword-based baseline system, we leverage SentencePiece (Kudo and Richardson, 2018) and train a 32k vocabulary on the training set. We also incorporate the 250k mBART-50 (Liu et al., 2020; Tang et al., 2020) vocabulary for full lexical coverage. For Byte-level systems, we preprocess data with a 256 vocabulary using scripts[2] from Shaham and Levy (2021).

Multilingual English-Centric Translation

We use the OPUS-7 corpus processed by Gu and Feng (2022), which is extracted from the OPUS-100 corpus (Zhang et al., 2020). The OPUS-7 dataset contains a training corpus of 6 languages (Ar, De, Fr, Nl, Ru, Zh) and their English Translations, with 1M sentences of each language.

Zero-shot Cross-domain Adaptation

Besides multilingual scenarios, we also experiment with the zero-shot cross-domain adaptation ability of byte-based translation models with the WMT19 German→English (De→En) dataset. We train all models on the News domain and evaluate on test data from three domains used in Sreedhar et al. (2023) and Aharoni and Goldberg (2020) , which are Koran, IT, and Medical. We use the data preprocessed and provided by Sreedhar et al. (2023).

3.2 Models

We compare the proposed MSC approach mainly with other byte-based machine translation models.

- **Transformer** (Vaswani et al., 2017): The standard Transformer model without adaptations to byte sequences.
- **Byte-*n*CF** (Sreedhar et al., 2023): A strong byte-based model performing well under low-resource settings. The structure hyperparameters are of default setting[3].
- **MSC**: We set n=8 in our experiments. The selection of k is discussed in Appendix A.

We also compare with subword-based methods.

- **Learned**: The standard Transformer model with a learned vocabulary.

[2] https://github.com/UriSha/EmbeddinglessNMT/blob/master/embeddingless_scripts/byte_preprocess.sh

[3] https://github.com/makeshn/LOBEF_Byte_NMT/blob/main/embeddingless_scripts/train_byte_ncf.sh

Approach	Param.	Direction	Ar	De	Fr	Nl	Ru	Zh	AVG	AVG-all
Transformer-subword	60.5M	XX→ En	36.60	34.02	34.10	30.28	36.77	38.82	35.30	31.27
		En→ XX	21.61	29.66	31.84	27.97	30.70	25.72	27.92	
Transformer-byte	44.6M	XX→ En	$28.79_{(\pm 0.30)}$	$29.68_{(\pm 0.11)}$	$27.75_{(\pm 0.21)}$	$26.58_{(\pm 0.04)}$	$27.66_{(\pm 0.30)}$	$27.94_{(\pm 0.25)}$	$28.06_{(\pm 0.14)}$	$25.19_{(\pm 0.07)}$
		En→ XX	$13.51_{(\pm 0.33)}$	$26.45_{(\pm 0.23)}$	$25.09_{(\pm 0.09)}$	$24.00_{(\pm 0.12)}$	$18.28_{(\pm 0.23)}$	$26.59_{(\pm 0.20)}$	$22.32_{(\pm 0.03)}$	
Byte-nCF	47.0M	XX→ En	$\mathbf{31.19}_{(\pm 0.31)}$	$\mathbf{30.86}_{(\pm 0.33)}$	$29.28_{(\pm 0.28)}$	$27.76_{(\pm 0.20)}$	$29.17_{(\pm 0.23)}$	$29.65_{(\pm 0.17)}$	$29.65_{(\pm 0.22)}$	$26.59_{(\pm 0.19)}$
		En→ XX	$14.58_{(\pm 0.14)}$	$27.58_{(\pm 0.21)}$	$26.20_{(\pm 0.14)}$	$25.05_{(\pm 0.22)}$	$\mathbf{19.60}_{(\pm 0.51)}$	$28.17_{(\pm 0.20)}$	$23.53_{(\pm 0.17)}$	
MSC	44.8M	XX→ En	$31.16_{(\pm 0.11)}$	$\mathbf{30.86}_{(\pm 0.22)}$	$\mathbf{29.31}_{(\pm 0.06)}$	$\mathbf{28.10}_{(\pm 0.13)}$	$\mathbf{29.49}_{(\pm 0.15)}$	$\mathbf{29.75}_{(\pm 0.18)}$	$\mathbf{29.78}_{(\pm 0.05)}$	$\mathbf{26.79}_{(\pm 0.06)}$
		En→ XX	$\mathbf{14.86}_{(\pm 0.12)}$	$\mathbf{27.89}_{(\pm 0.47)}$	$\mathbf{26.62}_{(\pm 0.15)}$	$\mathbf{25.39}_{(\pm 0.23)}$	$19.57_{(\pm 0.18)}$	$\mathbf{28.43}_{(\pm 0.34)}$	$\mathbf{23.80}_{(\pm 0.16)}$	

Table 2: The experiment results on OPUS-7 dataset, measured by SacreBLEU. All approaches are trained on the 12 directions together. Byte-based models are experimented three times and the table shows the average scores and the standard deviation.

- **mBART**: The standard Transformer model using the vocabulary of mBART (Liu et al., 2020). We do not use the pretrained checkpoint of mBART for fairness.

- **Aharoni**: The strongest baseline of subword-based models in this parameter scale.

The other settings are discussed in Appendix B.

4 Results and Analyses

4.1 Multilingual Many-to-One Translation

Table 1 shows the results on TED-59 datasets. All byte-based methods are experimented three times to enhance the reliability of results. We report the average and standard deviation of the results. Aharoni et al. (2019) have selected four low-resource (LR) and four high-resource (HR) languages to show models' performance on different training data scales, and we report results in the same way.

The average SacreBLEU scores of 58 translation directions (AVG-58) demonstrate that byte-based models are superior to subword-based models in massively multilingual scenario, despite of lower parameter usage.

Compared with other byte-based approaches, MSC performs better in almost all languages. While Byte-*n*CF learns a fixed set of combination weights of multi-scale contextual information for all languages, MSC adaptively leverages contextual information of different granularities at inference stage. For example, a single byte can represent a character or even a word in German, Italian, etc., so MSC leverages contextual information from its nearer neighborhood; a single byte may not be sufficient to form even a character, so MSC inclines to focus on contextual information of larger scales. We demonstrate this explanation later with an experiment in 4.4.

4.2 Multilingual English-Centric Translation

Table 2 shows the results on OPUS-7 dataset, which contains only seven high-resource languages. In this scenario, the subword-based model largely surpasses byte-based models. However, the performance gap is smaller when measured by COMET (Rei et al., 2022), a more reliable model-based metric, as reported in Appendix C.

Among these byte-based models, MSC generally performs better than the others. To verify such improvements are not from randomness, we repeat them for three times and report the average and standard deviation of the results.

4.3 Zero-Shot Cross-Domain Adaptation

Table 3 shows the results on in-domain and zero-shot out-of-domain test datasets. For subword-based models, using the dictionary trained on the News domain dataset to preprocess test sets from other domains results in a significant number of <*unk*> words. This leads to the model struggling to comprehend the input sentences and perform translations. However, byte-based models can eliminate the Out-Of-Vocabulary (OOV) issue, thereby achieving better performance in zero-shot translation scenarios. The results also demonstrate that our proposed MSC method has a significant advantage in zero-shot cross-domain adaptation.

4.4 Contextualization Scales

In section 2, we have introduced the hyper-parameter k which controls the contextualization scale of our approach. Here, we experiment on TED-59 dataset to show how this modeling scale affect translation.

According to the Unicode rule, we group languages by the number of bytes they require to form a character, which are named "Byte-1", "Byte-2", and "Byte-3". Then, we select three languages for each group to represent that group, as listed below.

Domain	Transformer-subword* (68.7M)	Transformer-byte* (44.3M)	Byte-nCF* (46.7M)	Transformer-subword (64.6M)	Transformer-byte (44.2M)	Byte-nCF (46.8M)	MSC (44.7M)
News	17.6	21.2	21.3	21.06	21.58	21.81	**21.86**
Koran	1.8	6.6	**7.4**	1.46	6.74	6.58	6.83
IT	3.9	10.4	11.6	2.73	10.89	11.33	**12.49**$^{\Uparrow}$
Medical	4.1	13.6	15.3	2.79	17.19	17.41	**20.01**$^{\Uparrow}$
AVG	3.27	10.20	11.43	2.33	11.61	11.77	**13.11**

Table 3: The experiment results on WMT19 De→En domain adaptation dataset. The "*" sign denotes the results are cited from Sreedhar et al. (2023). The average results of Koran, IT, and Medical domains indicate byte-based models perform better than subword-based models when test sets contain many rare words or even unknown words. The "⇑" sign denotes MSC prominently outperforms the second best method, with "$p < 0.001$".

- Byte-1: French (Fr), German (De), Dutch (Nl)
- Byte-2: Russian (Ru), Thai (Th), Arabic (Ar)
- Byte-3: Chinese (Zh), Japanese (Ja), Korean (Ko)

For the selection of k series, which reflect the contextualization scales applied in a model, we experiment the **small** scales "0,0,1,1,3,3,5,5", the **large** scales "0,0,1,1,5,5,7,7", and the **balanced** scales "0,0,1,1,3,5,5,7". These models can leverage information of granularities of "1,3,5", "1,5,7", and "1,3,5,7" respectively[4].

	small	large	balanced
Byte-1	30.190*	29.941	**30.219**
Byte-2	21.251	**21.601***	21.545
Byte-3	10.302	10.686*	**10.712**

Table 4: The selection of hyper-parameter k series affects model performance of different language groups.

The results are exhibited in Table 4. First, the model of balanced scales performs the best averagely, because it is provided with contextualized information of more scales and has more options to choose. Second, if we ignore the balanced one and compare the other models, the performance is related to the language groups. The "*" sign indicates the better performance between two models. For "Byte-1" group, a small scale is sufficient to model certain semantics, so the smaller scaled model performs better. For "Byte-3" groups, it requires larger contextual scales to form certain meanings, so the larger scaled model performs better.

[4]We have also experimented the granularities "9", "11" and "13", but they are harmful to translation quality. It demonstrates the model is unable to process information from too many tokens well.

These discoveries shade light on the selection of k series, which is the k series should be compatible with the language of input text.

5 Conclusions

In this paper, we make two primary contributions. Firstly, we show when byte-based models outperforms subword-based models and when they don't. In massively multilingual translation scenarios which involves a wide range of languages, byte-based models exhibit clear superiority, particularly for low-resource languages. In limited language numbers with sufficient training data, byte-based models lags behind subword-based models. Secondly, we introduce a Multi-Scale Contextualization (MSC) method which enhances the adaptability of byte-based models to diverse inputs. By dynamically integrating multi-scale contextualized information, MSC outperforms other byte-based models in generally all languages.

Limitations

While our approach can adaptively integrate multi-scale contextualized information, all contextualizing scopes k are predetermined. We will explore fully adaptive methods for multi-scale information extraction and integration for future work.

Acknowledgements

We thank all the anonymous reviewers for their insightful and valuable comments. This paper is supported by National Natural Science Foundation of China (Grant No.62376260).

References

Roee Aharoni and Yoav Goldberg. 2020. Unsupervised domain clusters in pretrained language models. In

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 7747–7763, Online. Association for Computational Linguistics.

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.

Lukas Edman, Antonio Toral, and Gertjan van Noord. 2023. Are character-level translations worth the wait? an extensive comparison of character- and subword-level models for machine translation. *CoRR*, abs/2302.14220.

Shuhao Gu and Yang Feng. 2022. Improving zero-shot multilingual translation with universal representations and cross-mapping. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6492–6504, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Langlin Huang, Shuhao Gu, Zhuocheng Zhang, and Yang Feng. 2023. Enhancing neural machine translation with semantic units. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*.

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.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

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 NAACL-HLT 2019: Demonstrations*.

Ye Qi, Devendra Sachan, Matthieu Felix, Sarguna Padmanabhan, and Graham Neubig. 2018. When and why are pre-trained word embeddings useful for neural machine translation? In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 529–535, New Orleans, Louisiana. Association for Computational Linguistics.

Ricardo Rei, José G. C. de Souza, Duarte Alves, Chrysoula Zerva, Ana C Farinha, Taisiya Glushkova, Alon Lavie, Luisa Coheur, and André F. T. Martins. 2022. COMET-22: Unbabel-IST 2022 submission for the metrics shared task. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 578–585, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Elizabeth Salesky, Neha Verma, Philipp Koehn, and Matt Post. 2023. Multilingual pixel representations for translation and effective cross-lingual transfer. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13845–13861, Singapore. 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, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Uri Shaham and Omer Levy. 2021. Neural machine translation without embeddings. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 181–186, Online. Association for Computational Linguistics.

Makesh Narsimhan Sreedhar, Xiangpeng Wan, Yu Cheng, and Junjie Hu. 2023. Local byte fusion for neural machine translation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7199–7214, Toronto, Canada. Association for Computational Linguistics.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2020. Multilingual translation with extensible multilingual pretraining and finetuning. *CoRR*, abs/2008.00401.

Yi Tay, Vinh Q. Tran, Sebastian Ruder, Jai Prakash Gupta, Hyung Won Chung, Dara Bahri, Zhen Qin, Simon Baumgartner, Cong Yu, and Donald Metzler. 2022. Charformer: Fast character transformers via gradient-based subword tokenization. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

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, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Changhan Wang, Kyunghyun Cho, and Jiatao Gu. 2020. Neural machine translation with byte-level subwords. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 9154–9160. AAAI Press.

Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. 2022. ByT5: Towards a token-free future with pre-trained byte-to-byte models. *Transactions of the Association for Computational Linguistics*, 10:291–306.

Lili Yu, Daniel Simig, Colin Flaherty, Armen Aghajanyan, Luke Zettlemoyer, and Mike Lewis. 2023. MEGABYTE: predicting million-byte sequences with multiscale transformers. *CoRR*, abs/2305.07185.

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.

A Selection of k Series

The selection of k varied across different languages. A critical determinant is the number of UTF-8 bytes required to represent a character. For languages that use the Latin alphabet, where a single byte can represent a character, a smaller k suffices; conversely, for languages where a character corresponds to multiple bytes, the information density of a single byte is lower, necessitating a larger k. Additionally, we find the "PassThrough($\cdot$)" function indispensable as it preserves the original information in the hidden states.

Empirically, the k series is "0,0,1,1,3,3,5,5" for De→En domain adaptation dataset, "0,0,1,1,3,5,5,7" for OPUS-7 dataset, and "0,0,3,3,5,5,7,7" for TED-59 dataset.

B Detailed Model Settings

For a fair comparison, all model implementations are based on the Fairseq (Ott et al., 2019) codebase. In our experiments, all models contain 6 transformer encoder layers and 6 transformer decoder layers with 8 attention heads. The model dimension is 512 for word embedding and 2048 for feed-forward layers. The word embeddings of the encoder and decoder are shared for all models. For our method, the MSC module is applied to the first encoder layer. For Byte-*n*CF, we use the default model structure, but apply the shared-embedding setting, which on one hand prove to be better than its default settings, and on the other hand align with other experiments.

For model training, we train all models using 8 GPUs with batch size 8192 for each. We use adam optimizer (Kingma and Ba, 2015) with β=$(0.9, 0.98)$ and 4k warm-up steps. The peak learning rate is $5e$-4 in multilingual tasks and $7e$-4 in De→En cross-domain adaptation task. Besides, we apply dropout 0.1 and label smoothing 0.1 for all models. We apply an early stop of 10, and average the last 5 checkpoints for evaluation. All models are evaluated using the SacreBLEU score.

C Experiment Results Measured by COMET

LID	Transformer (44.3M)	Byte-nCF (47.0M)	MSC (45.0M)
az	$68.34_{(\pm 0.72)}$	$67.47_{(\pm 0.49)}$	$\mathbf{69.27}_{(\pm 0.42)}$
be	$70.52_{(\pm 0.80)}$	$70.06_{(\pm 0.27)}$	$\mathbf{71.33}_{(\pm 0.47)}$
gl	$78.00_{(\pm 0.17)}$	$78.04_{(\pm 0.20)}$	$\mathbf{78.61}_{(\pm 0.13)}$
sk	$76.49_{(\pm 0.18)}$	$76.39_{(\pm 0.46)}$	$\mathbf{77.39}_{(\pm 0.38)}$
AVG-LR	$73.34_{(\pm 0.38)}$	$72.99_{(\pm 0.35)}$	$\mathbf{74.15}_{(\pm 0.33)}$
ar	$74.13_{(\pm 0.51)}$	$74.00_{(\pm 0.20)}$	$\mathbf{74.74}_{(\pm 0.59)}$
de	$76.88_{(\pm 0.24)}$	$76.93_{(\pm 0.36)}$	$\mathbf{77.77}_{(\pm 0.27)}$
he	$75.82_{(\pm 0.49)}$	$75.65_{(\pm 0.27)}$	$\mathbf{76.63}_{(\pm 0.63)}$
it	$78.43_{(\pm 0.25)}$	$78.67_{(\pm 0.20)}$	$\mathbf{79.12}_{(\pm 0.16)}$
AVG-HR	$76.31_{(\pm 0.23)}$	$76.32_{(\pm 0.25)}$	$\mathbf{77.06}_{(\pm 0.41)}$
AVG-58	$73.00_{(\pm 0.37)}$	$74.65_{(\pm 0.30)}$	$\mathbf{75.61}_{(\pm 0.37)}$

Table 5: The experiment results on TED-59 dataset, measured by COMET.

Approach	Param.	Direction	Ar	De	Fr	Nl	Ru	Zh	AVG	AVG-all
Transformer-subword	60.5M	XX→ En	79.03	80.10	79.27	78.01	78.14	79.15	78.89	78.65
		En→ XX	78.50	78.41	76.94	78.05	79.09	79.72	78.45	
Transformer-byte	44.6M	XX→ En	$77.31_{(\pm 0.16)}$	$77.69_{(\pm 0.04)}$	$77.16_{(\pm 0.06)}$	$75.98_{(\pm 0.04)}$	$75.64_{(\pm 0.12)}$	$76.66_{(\pm 0.15)}$	$76.74_{(\pm 0.06)}$	$76.19_{(\pm 0.04)}$
		En→ XX	$75.00_{(\pm 0.10)}$	$74.37_{(\pm 0.02)}$	$72.51_{(\pm 0.05)}$	$74.11_{(\pm 0.06)}$	$71.38_{(\pm 0.05)}$	$86.46_{(\pm 0.08)}$	$75.64_{(\pm 0.02)}$	
Byte-nCF	47.0M	XX→ En	$78.68_{(\pm 0.25)}$	$78.62_{(\pm 0.10)}$	$78.24_{(\pm 0.20)}$	$77.18_{(\pm 0.31)}$	$76.77_{(\pm 0.17)}$	$77.88_{(\pm 0.42)}$	$77.90_{(\pm 0.23)}$	$77.55_{(\pm 0.26)}$
		En→ XX	$76.35_{(\pm 0.24)}$	$76.08_{(\pm 0.18)}$	$73.85_{(\pm 0.37)}$	$75.61_{(\pm 0.34)}$	$73.73_{(\pm 0.40)}$	$87.63_{(\pm 0.27)}$	$77.21_{(\pm 0.29)}$	
MSC	44.8M	XX→ En	$\mathbf{78.90}_{(\pm 0.06)}$	$\mathbf{78.84}_{(\pm 0.07)}$	$\mathbf{78.32}_{(\pm 0.05)}$	$\mathbf{77.37}_{(\pm 0.07)}$	$\mathbf{76.92}_{(\pm 0.13)}$	$\mathbf{78.09}_{(\pm 0.08)}$	$\mathbf{78.08}_{(\pm 0.06)}$	$\mathbf{77.85}_{(\pm 0.03)}$
		En→ XX	$\mathbf{76.77}_{(\pm 0.06)}$	$\mathbf{76.36}_{(\pm 0.10)}$	$\mathbf{74.32}_{(\pm 0.15)}$	$\mathbf{76.08}_{(\pm 0.03)}$	$\mathbf{74.28}_{(\pm 0.07)}$	$\mathbf{87.92}_{(\pm 0.17)}$	$\mathbf{77.62}_{(\pm 0.02)}$	

Table 6: The experiment results on OPUS-7 dataset, measured by COMET.