MoCE: Adaptive Mixture of Contextualization Experts for Byte-based Neural Machine Translation

Langlin Huang [1,3,4†], **Mengyu Bu** [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
[4] Washington University in St. Louis
h.langlin@wustl.edu, {bumengyu23z, fengyang}@ict.ac.cn

Abstract

Byte-based machine translation systems have shown significant potential in massively multilingual settings. Unicode encoding, which maps each character to specific byte(s), eliminates the emergence of unknown words, even in new languages. This avoids out-of-vocabulary risk in multilingual translation and enables broad language scalability. However, byte-level tokenization results in sequences that are hard to interpret due to limited semantic information per byte. Local contextualization has proven effective in assigning initial semantics to tokens, improving sentence comprehension. Nevertheless, variations in encoding rules across languages necessitate an adaptive approach for effective contextualization. To this end, we propose Mixture of Contextualization Experts (MoCE), adaptively selecting and mixing attention heads, which are treated as contextualization experts. This enhances the flexibility of contextualization scales and allows models to search for better contextualization combinations. Experiment results show that our method outperforms existing methods without extensive manual adjustment of hyper-parameters and surpasses subword-based models with fewer parameters in Ted-59 dataset. Our code is available at https://github.com/ictnlp/MoCE.

1 Introduction

Neural Machine Translation (NMT) is a consistently hot research topic, and recent years have seen the growing significance of multilingual language modeling (Zhang et al., 2023). The selection of tokenization and vocabulary is critical to multilingual language models, which plays an important role in vectorization of texts and discretization of predicted hidden states. While some models (Costa-jussà et al., 2022; Dubey et al., 2024) use large vocabularies to ensure word coverage, others (Touvron et al., 2023; Jiang et al., 2023) opt for byte fallback strategy. These approaches allow them to completely avoid unknown words with a smaller vocabulary size. Byte-based models like Xue et al. (2022); Yu et al. (2023); Shaham and Levy (2021) convert all words into UTF-8 byte, which further reduces the vocabulary size to about 256. This strategy also reduces the size of the embedding table, saving parameters and accelerating token embedding and inference. Besides, it eliminates the unknown-word problem and can be easily generalized to massively multilingual scenarios. Empirical study (Edman et al., 2024) has also shown the performance superiority of byte-based MNT models.

However, the drawbacks of byte-based models are obvious, most notably that an individual byte struggles to convey a specific semantic meaning. Therefore, various contextualization methods (Lee et al., 2017; Clark et al., 2022) have been proposed to alleviate this problem. MEGABYTE (Yu et al., 2023) reassembles byte streams into groups of four, constructing group representations by concatenating their hidden states. CharFormer (Tay et al., 2022) and LOBEF (Sreedhar et al., 2023) employ local-contextualization techniques to encode bytes, with CharFormer using mean-pooling and LOBEF using Convolutional Neural Networks (CNNs). MSC (Huang and Feng, 2024) argues that a byte should contribute to multiple neighboring contexts, necessitating a multi-scale contextualization approach. To this end, MSC groups hidden state dimensions and assigns CNNs with different kernel sizes to each group.

Although MSC provides an effective framework for modeling multi-scale contextualization and achieves state-of-the-art performance, it suffers from the limitation of manually predefined scales. This reduces the model's ability to generalize to multilingual scenarios, particularly in massively multilingual machine translation, which may in-

[*] Corresponding author.

[†] This paper was done when Langlin Huang studied at Institute of Computing Technology, Chinese Academy of Sciences.

Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 1011–1028
April 29 - May 4, 2025 ©2025 Association for Computational Linguistics

volve over 50 languages. Under UTF-8 rule, a character may convert to 1 to 4 bytes, depending on the language. This leads to varying requirements of contextualization scale for different languages. However, once MSC decides the contextualization scales, they are unchangeable for any input.

To address this, we leverage the concept of Mixture of Experts (MoE) (Shazeer et al., 2017) and propose Mixture of Contextualization Experts (MoCE), which can adaptively determine CNN kernel sizes based on each input text. Specifically, we modify Multi-Head Attention to propose Adaptive MultiScale-Headed Attention (Ada-MSHA) module. This proposed attention allows each head to be locally contextualized and the contextualization scales are adaptive to the input. Instead of predefined scales that MSC uses, MoCE dynamically combines different scales with model needs. The flexibility of contextualization scales is therefore significantly enhanced, resulting in better performance. Additionally, we prove that given language ID as prior knowledge benefits the scale selection.

Experiment results on two massively multilingual translation datasets, Ted-59 and OPUS-100, demonstrate our proposed method outperforms other byte-based translation models with similar parameter use. Compared with the subword-based model, MoCE requires fewer parameters while performing better in Ted-59 dataset.

2 Background

2.1 Mixture-of-Experts (MoE)

MoE was designed mainly to increase the potential parameters of a model (Shazeer et al., 2017; Lepikhin et al., 2021; Jiang et al., 2024). Recently, it is also used in multi-domain (Du et al., 2023) or multi-task (Park, 2024; Huang et al., 2023a) scenarios.

An "expert" usually represents a layer within the model, and MoE provides multiple counterparts for the same layer. During computation, only one or a small number of experts are activated at a time, increasing model's modeling ability with limited extra computational cost. It is worth noting that the counterparts are not necessarily the same structure; Ramachandran and Le (2019) uses heterogeneous experts, such as CNNs with different kernel sizes.

In terms of expert selection, a typical method is predicting the selection probability distribution of the experts and then choosing the k most possible experts (Shazeer et al., 2017; Ramachandran and Le, 2019). As shown in (1), x and y are the input and output of an MoE layer respectively. E_i represents the i^{th} expert, and G_i is the normalized probability of choosing the i^{th} expert, given by (2).

$$y = \sum_{i}^{n} G_i(x)E_i(x) \tag{1}$$

$$G(x) = \mathrm{Softmax}(\mathrm{Top_k}(P(x), k)) \tag{2}$$

where $\mathrm{Top_k}$ function is given by (3):

$$\mathrm{Top_k}(v_j, k) = \begin{cases} v_j, v_j \text{ in top k elements of } v \\ -\infty, \text{otherwise} \end{cases} \tag{3}$$

2.2 Multi-Scale Contextualization

In contrast to commonly used global contextualization, multi-scale contextualization is a local one. The nature of byte-based texts necessitates the involvement of local contextualization (Tay et al., 2022; Sreedhar et al., 2023). Then, MSC (Huang and Feng, 2024) extends this to contextualization of various scales, adapting to different contexts.

The multi-scale contextualization functions given by MSC are simple and direct:

$$g_i(\cdot, r) = \begin{cases} \mathrm{Identity}(\cdot) & , k = 0 \\ \mathrm{CNN}(\cdot, k) & , k > 0 \end{cases} \tag{4}$$

Where k denotes the contextualization realm, i.e. the kernel size of CNN. Empirically, k is chosen from 0, 1, 3, 5, 7 (Huang and Feng, 2024).

2.3 Multi-Head Attention (MHA)

MHA is one of the core components of Transformer structure (Vaswani et al., 2017), which employs multiple attention functions in parallel instead of a single one. Each attention function corresponds to an independent attention head, which linearly projects the input to Query, Key and Value vectors with a reduced dimension and applies "Scaled Dot-Product Attention" (Vaswani et al., 2017) on them. Eventually, outputs of all heads are concatenated to a full dimension.

An equivalent view of MHA is it breaks the hidden state dimensions of the linearly projected vectors into h parts, as shown in (5),

$$\begin{cases} Q = & XW^Q = & [Q_1, Q_2, ..., Q_h] \\ K = & XW^K = & [K_1, K_2, ..., K_h] \\ V = & XW^V = & [V_1, V_2, ..., V_h] \end{cases} \tag{5}$$

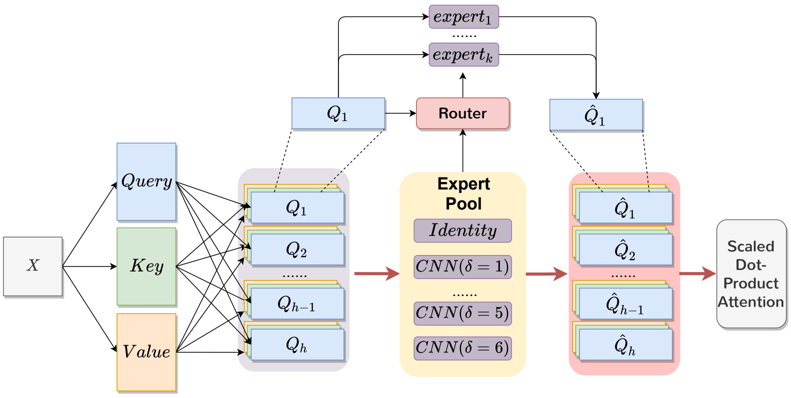


Figure 1: This figure shows the overall structure of MoCE. All contextualization functions are treated as experts. The router dynamically allocates experts from the expert pool with their corresponding weights for each head of each Q, K, and V vector. The heads are locally contextualized with given experts, and then serve as the input to the following Scaled Dot-Product Attention module.

and applies "Scaled Dot-Product Attention" for each, as shown in (6). Finally, MHA is given by (7).

$$\text{head}_i = \text{softmax}(\frac{Q_i K_i^T}{\sqrt{d_k}})V_i \tag{6}$$

$$\text{MHA}(X) = \text{Cat}(\text{head}_1, ..., \text{head}_h)W^O \tag{7}$$

2.4 Multilingual NMT

For many-to-many multilingual NMT, it requires language IDs to indicate the source and target languages (Johnson et al., 2017). A typical approach is prepending a language token at the beginning of the source and target sentences, respectively. Table 1 provides an example.

	Source	Target
Origin	Hello world!	Bonjour le monde!
Model Input	<en> _Hello _world !	<fr> _Bonjour _le _monde !

Table 1: An example of language token.

3 Method

In this section, we first introduce MultiScale-Headed Attention, a way to model multi-scale contextualization. Then we extend it to Adaptive MultiScale-Headed Attention, the core module of MoCE. We also propose how to leverage language ID information to further improve our method.

3.1 MultiScale-Headed Attention (MSHA)

Grouping hidden state dimensions has been a common and effective way to perform different operations on the same token (Huang and Feng, 2024; Wu et al., 2024). Since it is the same as a part of MHA, we take advantage of MHA and use its heads as the grouped dimensions, like in (5).

To perform multi-scale contextualization on each head, MSHA applies the same contextualization function $g(\cdot)$ on Q, K, and V vectors. Taking the i^{th} head as an example, it substitutes the vectors in (6) with contextualized form, yielding (8).

$$\widetilde{\text{head}_i} = \text{softmax}(\frac{g_i(Q_i)g_i(K_i)^T}{\sqrt{d_k}})g_i(V_i) \tag{8}$$

For $g(\cdot)$, we rewrite (4) in a clear way as in (9). Here δ denotes the neighborhood radius (including the central word) of local contextualization.

$$g(\cdot,\delta) = \begin{cases} \text{Identity}(\cdot) & ,\delta = 0 \\ \text{CNN}(\cdot, 2\delta - 1) & ,\delta > 0 \end{cases} \tag{9}$$

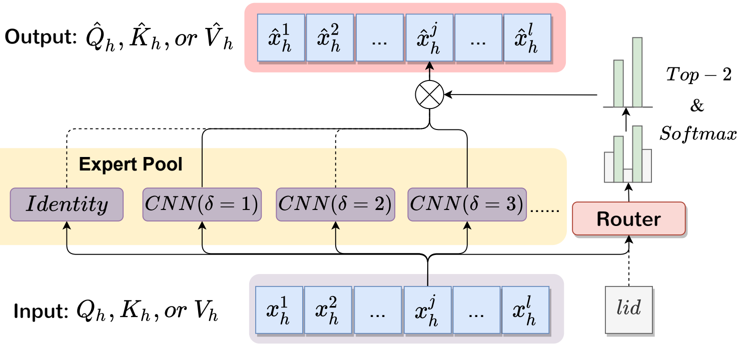
Similarly, we replace head_i from (8) with $\widetilde{\text{head}_i}$.

$$\text{MSHA}(X) = \text{Cat}(\widetilde{\text{head}_1}, ..., \widetilde{\text{head}_h})W^O \tag{10}$$

MSHA offers two advantages over MSC. First, it leverages the natural grouping operation within MHA, saving an additional vector separation and recombination step. Second, each head in MSHA is responsible for contextualization at a specific granularity, enhancing the model's interpretability, facilitating analysis presented in Section 5.4.

3.2 Adaptive MultiScale-Headed Attention

Based on MSHA, we leverage MoE structure to propose an adaptive approach, Ada-MSHA, to



Output: $\hat{Q}_h, \hat{K}_h, or\ \hat{V}_h$
$\hat{x}_h^1$ $\hat{x}_h^2$ ... $\hat{x}_h^j$ ... $\hat{x}_h^l$
$Top-2$
&
$Softmax$
Expert Pool
$Identity$
$CNN(\delta=1)$
$CNN(\delta=2)$
$CNN(\delta=3)$
Router
Input: $Q_h, K_h, or\ V_h$
x_h^1 x_h^2 ... x_h^j ... x_h^l
lid

Figure 2: This figure shows the detailed model structure around the routing mechanism. The input x_h denotes the h^{th} head of Q, K, or V vector, with the sentence length of l. x_h^j represents the j^{th} token of x_h. For an arbitrary token x_h^j, the router predicts the selection probability for each expert and selects the top-2. The weighted combination of contextualized vectors from these 2 experts forms the final output $\hat{x}_h^j$. "lid" represents the language token, which is optional prior information for the router. If applied, "lid" is concatenated with x_h^j to be the input.

solve the problem of fixed contextualization scales. In our approach, contextualization functions $g(\cdot)$ are not determined by hyper-parameters. Rather, the model decides which $g(\cdot)$ to use. Specifically, a router predicts the selection probability of the candidate $g(\cdot)$ for each token respectively, and it selects $g(\cdot)$ according to the predicted probabilities. Viewing Ada-MSHA from the perspective of MoE, different contextualization functions serve as experts and they compose an expert pool, which the router selects experts from. The overall model structure is depicted in Figure 1.

As for routing mechanism, Ada-MSHA basically applies the commonly used implementation (Shazeer et al., 2017; Ramachandran and Le, 2019) for routing, i.e. (1), (2), and (3).

Figure 2 shows the routing mechanism with x_h^j as an example. The input x can be any head of Q, K, or V vectors. x_h^j means the h^{th} head from the j^{th} token of the sentence x. On one side, the router takes x_h^j as input and predicts the selection probability for all experts. Then, the top-k, where k is default as 2, probabilities are normalized to be a distribution. On the other side, the corresponding k experts take x_h^j as input and output the contextualized vector, $g_i(x_h^j)$. The final output $\hat{x}_h^j$ is given by the weighted summation of these k vectors, as shown in (11), which aligns with (1).

$$\hat{x}_h^j = \sum_i^n G_i(x_h^j) g_i(x_h^j) \tag{11}$$

The candidate contextualization functions, $g(\cdot)$, are manually defined in MSC (Huang and Feng, 2024), which has demonstrated it beneficial to provide $g(\cdot)$ of more scales. Therefore, we provide contextualization functions with various δs, consecutively from 0 to Δ, the predefined upper bound. For example, if $\Delta = 5$, there are 6 $g(\cdot)$, with $\delta = 0, 1, 2, 3, 4, 5$ for each. In this way, we reduce the number of hyper-parameters from 8 in MSC to only 1 in Ada-MSHA.

It should be clarified that Ada-MSHA is a modified Attention layer. By default, we replace the first encoder layer with Ada-MSHA and keep the rest model parts unchanged. This is discussed in Appendix G.

3.3 Language ID - the Prior Information

The standard $P(x)$ from (2) is given by (12), where W^R stands for the router and is a linear mapping from hidden state dimension to the number of expert candidates.

$$P(x) = \text{softmax}(xW^R) \tag{12}$$

Realizing a byte may be interpreted differently as the language changes, we propose to concatenate the language ID (lid) token with x to serve as router's input. The "$+lid$" version router is given by (13), where "$[\cdot|\cdot]$" denotes concatenation.

$$P(x) = \text{softmax}([x|lid]W^R) \tag{13}$$

The experiment results in I demonstrate the "lid" is beneficial to massively multilingual translation.

4 Experiment Settings

To verify the effectiveness of our proposed approach, we experiment on two massively multilingual translation datasets with over 50 languages.

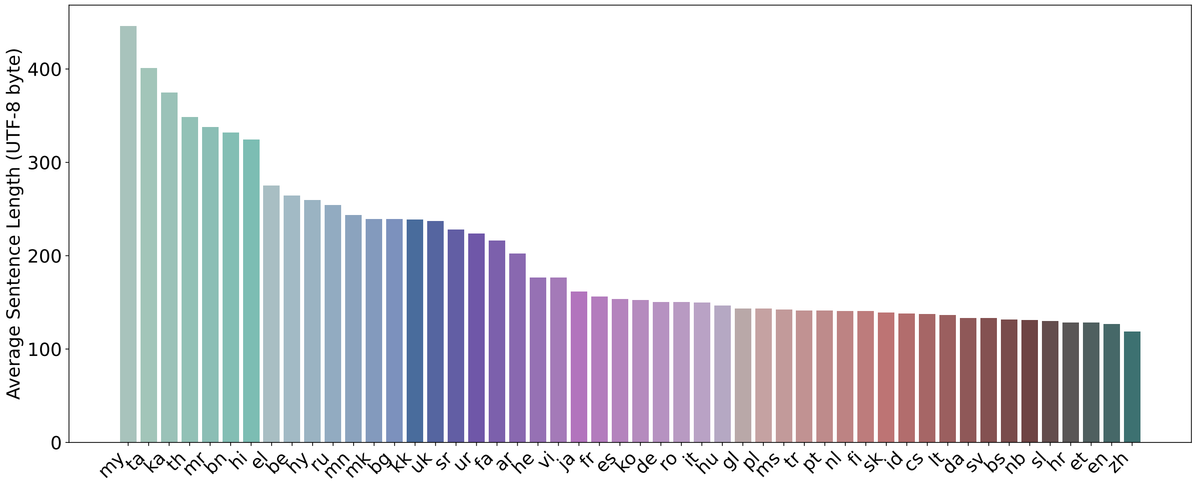


Figure 3: This figure depicts the average length of byte-based strings of different languages to express the same meaning. It reflects the complexity/conciseness of languages; languages are getting more concise from left to right.

We mainly compare with byte-based models but also try the subword-based Transformer model.

4.1 Datasets

Ted-59

Following Huang and Feng (2024), we use Ted-59 dataset (Qi et al., 2018) for fair comparison. This dataset stems from the TED corpus and comprises 59 languages, including English and 58 other languages. All sentence pairs are English-centered. We use the data provided by Salesky et al. (2023).

For data preprocessing, we apply the scripts from EmbeddinglessNMT (Shaham and Levy, 2021) to perform byte-level preprocessing. The final vocabulary contains 256 bytes as well as several language tokens. For the subword-based system, SentencePiece (Kudo and Richardson, 2018) with vocabulary size 32k is applied for tokenization.

OPUS-100

OPUS-100 (Zhang et al., 2020) is an even larger massively multilingual translation dataset, which covers English and 99 other languages.

The data preprocessing is the same as that for Ted-59, except using Sentencepiece with vocabulary size 64k, aligning with Zhang et al. (2020).

4.2 Systems

We compare our method with the following systems on multilingual benchmarks:

- Transformer (Vaswani et al., 2017): Baseline that can be byte-based or subword-based.
- Byte-nCF (Sreedhar et al., 2023): A strong byte-based NMT model, with default hyper-parameter settings[1].
- MSC (Huang and Feng, 2024): The state-of-the-art byte-based NMT model prior to Ada-MSHA. For Ted-59, we use recommended hyper-parameters[2]. For others, we use more contextualization scales which proves better[3].
- MoCE: Our proposed method, with hyper-parameter $\Delta = 5$ or 6. Routing with language ID is also tested, shown as "$+lid$".

Category	Languages
Long	my, ta, ka, th
Medium	bg, mk, uk, sr
Short	eo, sl, sv, et
L.R.	az, be, gl, sk
H.R.	ar, de, he, it
OPUS4	de, zh, br, te

Table 2: Language composition for each category.

4.3 Training, Inference, and Evaluation

We follow the standard practice in multilingual translation by training models on both "*xx*→*en*" and "*en*→*xx*" translation directions. Since the focus is handling diverse source languages, our evaluation primarily targets the "*xx*→*en*" direction. Apart

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

[2] $k = 0, 0, 3, 3, 5, 5, 7, 7$ in (4)

[3] $k = 0, 0, 1, 1, 3, 5, 5, 7$ in (4)

	Param.	Long	Medium	Short	L.R.	H.R.	All
Transformer (Subword)	60.6M	15.43	31.05	26.34	23.57	30.69	24.79
Transformer (Byte)	44.3M	14.66	32.06	27.43	25.01	31.91	25.21
Byte-*n*CF	46.7M	13.75	31.09	26.31	23.55	30.84	24.33
MSC	44.4M	14.86	32.36	28.00	25.11	32.26	25.61
MoCE ($\Delta = 5$)	44.4M	15.89	**33.23**	28.56	25.81	32.92	26.30
$+lid$	44.4M	16.28	33.19	28.64	**25.84**	**33.02**	**26.52**
MoCE ($\Delta = 6$)	44.5M	15.91	32.59	28.44	25.34	32.65	26.13
$+lid$	44.5M	**16.42**	32.89	**28.77**	25.49	**33.04**	26.43

Table 3: Overall BLEU scores on Ted-59 dataset. The definition of each category is detailed in section 5.2. "All" means the average score of all 58 translation directions.

from reporting BLEU scores in the main body, we also report character-level metric ChrF and model-based metric COMET in Appendix E and D for comprehensive evaluation. We detail the complete setups in Appendix B.

5 Results and Analysis

5.1 Language Conciseness under UTF-8 Byte Encoding

Before translation experiments, we first examine the conciseness of different languages. As previously mentioned, a character may be represented by 1 to 4 UTF-8 bytes. A misleading intuition is that languages using 3-byte characters are longer in sentence lengths. However, languages like Chinese, are inherently concise, which is easily overlooked.

To explore language conciseness under byte encoding, we use Flores-101 dataset (Goyal et al., 2022), which contains parallel sentence pairs across 101 languages, ensuring that all languages convey the same semantic content. Inspired by Limisiewicz et al. (2024), we calculate the average length of byte-encoded sentences for each language, which reflects the conciseness of languages. The results are shown in Figure 3. The experiment details are introduced in Appendix A.

5.2 Language Categorization

To effectively compare and analyze the results of multilingual translation, we report scores across different language categories. For each category, we select four representative languages and present their average scores. We first group languages into three categories: "Long", "Medium", and "Short", based on the average sentence length in section 5.1. Following Huang and Feng (2024), languages are grouped into low-resource (L.R.) and high-resource (H.R.) categories based on corpus size. Following Zhang et al. (2020), four languages as selected as "OPUS4". The compositions of all categories are shown in Table 2.

5.3 Main Experiments

Results on Ted-59

Table 3 summarizes the results on Ted-59 dataset. Nearly all byte-based models outperform the subword-based model with much fewer parameters. Among the byte-based models, all varieties of MoCE surpass MSC by a large margin, demonstrating the effectiveness of using adaptive contextualization scales.

The comparison of different settings of MoCE demonstrates two points. First, applying "$+lid$" achieves consistent improvement, which is discussed in Appendix I. Second, Δ value affects model's inclination toward language groups, especially "Long" and "Medium" groups, which is discussed in Section 5.5. The results at "Short" group do not follow a certain rule. We conjecture this is because a single byte has a determined mapping to character for "Short" languages, so they rely less on contextualization functions.

Results on OPUS-100

Table 4 exhibits the results on OPUS-100 dataset.[4] The "*" denotes the results from Zhang et al. (2020), while our re-implementation is much better than the original one.

The comparison between subword-based and byte-based models differs from Ted-59. The reason lies in two sides. First, the subword-based model doubles vocabulary size, reducing the risk of encountering unknown words. Second, OPUS-100

[4] Byte-*n*CF (Sreedhar et al., 2023) is absent here because the source code fails to support such massive data volume, despite we used up our 300GB memory.

	Param.	Long	Medium	Short	OPUS4	All
Transformer* (Subword)	110M	-	-	-	23.35	27.60
Transformer (Subword)	77.0M	20.72	28.07	30.43	29.41	30.72
Transformer (Byte)	44.3M	16.63	24.47	24.84	22.81	25.36
MSC	44.4M	16.38	24.79	25.48	23.65	25.74
MoCE ($\Delta = 5$)	44.4M	16.44	24.55	25.14	23.16	25.31
+*lid*	44.4M	16.48	**25.03**	25.68	23.13	25.79
MoCE ($\Delta = 6$)	44.5M	16.48	24.92	25.47	23.58	25.79
+*lid*	44.5M	**17.06**	24.94	**25.93**	**24.01**	**26.10**

Table 4: Overall BLEU scores on OPUS-100 dataset. The definition of each category is detailed in section 5.2. "All" means the average score of all 94 translation directions. We bold the highest BLEU scores for byte-based methods.

provides more training data, allowing effective convergence of the expanded embedding table.

The previous conclusion that Δ influences models' inclination toward different groups is less obvious also due to the sufficiency of training data. A larger Δ provides more expert choices, but a well-converged router can avoid choosing the inappropriate experts. As a result, $\Delta = 6$ performs no worse than $\Delta = 5$ in all groups.

5.4 Expert Choice and Language Conciseness

The major motivation of MoCE is to choose appropriate contextualization functions according to the input. To verify if MoCE achieves such target, we count the selected ratio of each expert throughout a whole test set. Specifically, we select 4 languages, *my*, *bg*, *et*, and *zh*, from left to right in Figure 3, and also from least concise to most concise. Then, we test "*en*→*xx*" and "*xx*→*en*" for them, recording the selected ratios of experts. The sentence length ratios xx/en are recorded to represent the relative conciseness, which highly correlates to the expert choices. All experiments are conducted on Ted-59 with "+*lid*" setting.

The results are exhibited in Figure 4, where four columns represent different languages ("*xx*"), and two rows represent model settings ($\Delta = 5$ and 6). The sentence length ratios of "*xx*" over "*en*" are listed in the middle line. For all the sub-figures, the solid lines ("*en*→*xx*") are treated as pivots, because they share the same source language. The differences between "*xx*→*en*" and the pivots reveal model's inclination towards different languages.

Taking either row as an example, the model gradually tends to choose smaller contextualization radius (δ) as "*xx*" becomes more concise. To quantitatively show this δ-shift behavior, we drew the averages of δ in vertical lines. For the least concise language, "*my*", model tends to choose larger δ than the pivot. For the most concise language, "*zh*", model tends to choose smaller δ than the pivot. To measure the differences in more detail, we calculated the JS-divergence between the distributions of "*xx*→*en*" and "*en*→*xx*". The combination of figures and JS-divergence values clearly show the decreasing tendency of selected ratios for experts with larger δ and the increasing tendency of those with smaller δ, from left to right.

These findings demonstrate that MoCE can effectively identify the input language type and route the input to appropriate experts.

5.5 Systematic Shift of δ with Varied Δ

Apart from the input language type, the choice of δ is also influenced by the upper bound of contextualization radius, which is Δ. Intuitively, providing experts with larger δ encourages a model to leverage longer context information, though this is not always more beneficial for the overall performance.

Comparing $\Delta = 5$ and 6 in Figure 4, we observe a consistent shift of δ selection. This shift of overall tendency is systematic, caused by model structure change. The vertical lines quantitatively demonstrate a model with larger Δ tends to select experts with a larger radius. This explains why $\Delta = 6$ performs better in "Long" and $\Delta = 5$ performs better in "Medium".

5.6 How Δ Influences Translation Quality

After analyzing how Δ influences expert choices, we empirically assess its influence on translation quality. Specifically, we conduct experiments on Ted-59 with different Δ, and focus on different sentence length groups. In addition to previous experiments, we try a smaller $\Delta = 4$ for comparison.

Figure 5 exhibits their advantages over Transformer (Byte). While $\Delta = 6$ performs the best in

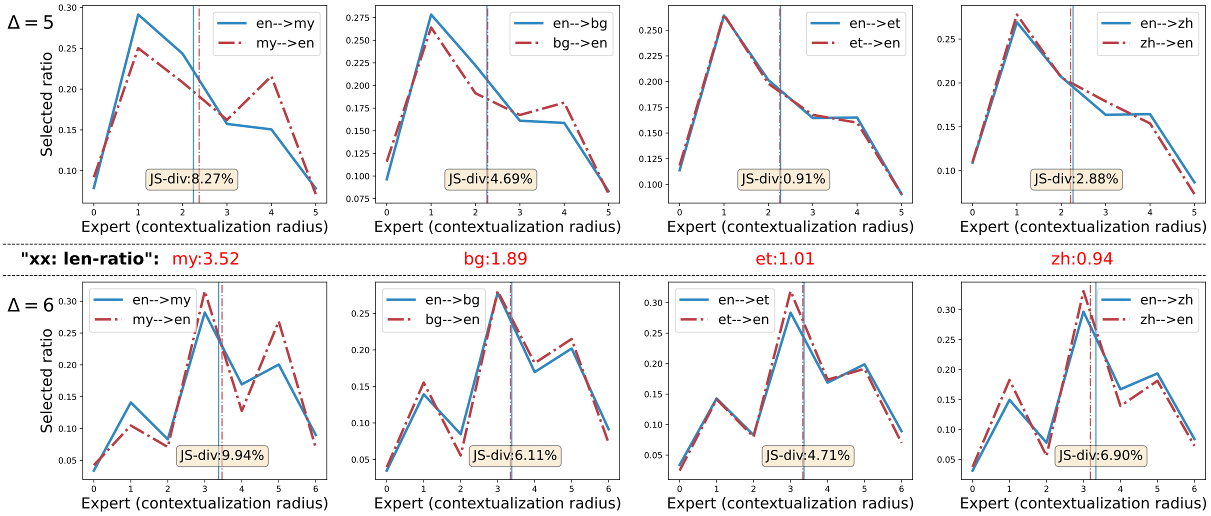


Figure 4: These figures exhibit the selected ratio of each expert, recorded when translating "$en{\rightarrow}xx$" and "$xx{\rightarrow}en$". The vertical lines are the averaged contextualization radius (δ). The two rows represent $\Delta = 5$ and 6 respectively. Listed in the middle line are what "xx" stands for and its length ratio over "en", reflecting the relative conciseness. **From left to right**, as the language conciseness for "xx" increases, the router tends to select more experts with smaller δ. The JS-divergences between two distributions quantitatively prove it too. This consistency demonstrates that MoCE has learned to select proper experts for corresponding input. **From top to bottom**, as Δ increases, the model systematically tends to choose larger δ. This explains why "$\Delta = 6$" is better in "Long" and "$\Delta = 5$" is better in "Medium" in Table 3.

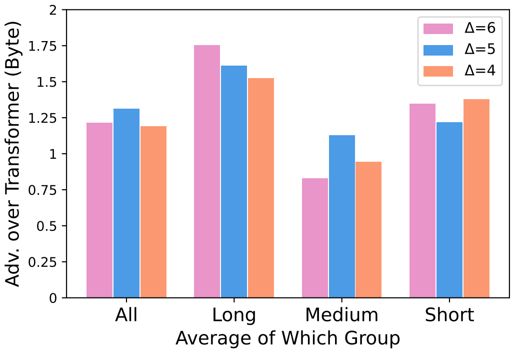


Figure 5: This figure compares MoCE with different Δ and show their advantage over Transformer (Byte). BLEU is used in this figure. Their performance gaps in different language groups show that Δ benefits languages with corresponding conciseness.

"Long" group, $\Delta = 5$ and 4 perform the best in "Medium" group and "Short" group respectively. This figure demonstrates that for a certain language, using the appropriate contextualization function can maximize translation quality.

5.7 Effectiveness of MoCE Concept

We highlight the benefit of applying a Mixture of Contextualization Experts over using a single expert for each head. With the weights of experts varying in continuous space, the defacto contextualization radius has way more possibilities. Here, top-1 routing is chosen to simulate the single expert setting. Empirical results in Table 5 demonstrate the advantage of applying MoE in our method. It also proves that mixing two experts is enough.

	Long	Medium	Short	Avg.
Top-1	15.86	32.74	28.29	26.03
Top-2	**16.28**	**33.19**	**28.64**	**26.52**
Top-3	16.15	32.90	28.51	26.34

Table 5: Top-2 routing is the standard MoCE, and Top-1 simulates the strategy of not mixing experts. Experiments are conducted with "$\Delta = 5$, $+lid$" on Ted-59.

5.8 Training and Inference Speed

While MoCE demonstrates substantial improvements in translation quality over the baselines, the concern of time cost may rise. To address this concern, we evaluated the training and inference speed of all methods mentioned in this paper using Ted-59 dataset, providing a quantitative comparison of computational efficiency. The byte-based Transformer model serves as the baseline, the relative speed ratios are exhibited in Table 6.

For the training phase, we recorded the time spent in training one epoch. For the inference phase, we recorded the time spent in translating the entire test set for each translation direction, and

	Training	Inference
Transformer (byte)	1x	1x
MSC	0.96x	0.96x
Byte-nCF	0.95x	0.97x
MoCE ($\Delta = 5$)	0.98x	0.96x
+lid	0.99x	0.96x
MoCE ($\Delta = 6$)	0.96x	0.96x
+lid	0.98x	0.96x

Table 6: The relative speed ratio compared with byte-based Transformer baseline, measured on Ted-59 dataset.

took the sum of them. The speed of Transformer (byte) is denoted as "1x"; values below 1 indicate slower speeds compared to the baseline. According to Table 6, all the local contextualization methods, including MoCE, achieve nearly the same speed as the baseline.

6 Related Works

6.1 Byte-based Multilingual NMT

Most multilingual NMT systems utilize subword-level tokenization techniques such as Byte Pair Encoding (Sennrich et al., 2016) and Unigram (Kudo, 2018). They often result in over-segmentation of low-resource languages, issues with out-of-vocabulary (OOV) words, and difficulties in adapting to new languages or domains (Rust et al., 2021; Raunak et al., 2020). As the number of languages increases, these problems worsen, limiting the performance of multilingual NMT systems. To this end, various byte-level approaches have been proposed. Shaham and Levy (2021) build embedding-free NMT models using UTF-8 byte tokenization. Tay et al. (2022) propose CharFormer, integrating byte representations from different block sizes by mean-pooling. Sreedhar et al. (2023) enhance CharFormer by employing CNN for representation integration. Huang and Feng (2024) propose MSC, which assigns contextualized information at different scales across various hidden state dimensions.

6.2 MoE for Multilingual NMT

Previous studies on multilingual NMT have demonstrated that model performance initially improves but subsequently declines as the number of training languages and the corpus size increase (Arivazhagan et al., 2019; Zhang et al., 2020; Yang et al., 2021). This phenomenon, known as the "curse of multilingualism", is attributed to limited model capacity (Chang et al., 2023). The MoE techniques significantly scale up model capacity without increasing training and inference costs accordingly, and can be used to scale up multilingual NMT models and mitigate language conflicts. Shazeer et al. (2017) propose sparse-gated modules with thousands of experts and validate the effect on bilingual translation tasks. Li et al. (2023) use pre-trained FFN to initialize each expert, and the model can flexibly combine dense and sparse MoE modules. To address the problem of insufficient expert training in sparse MoE, Wu et al. (2024) integrates the knowledge of experts through token-splitting-merging operation, which significantly improves the expert activation ratio.

6.3 Local Contextualization Modeling

While the attention mechanism's global contextualization effectively addresses the challenge of long-distance dependencies, many tasks benefit from a focus on local contextualization. For instance, in audio processing, a model needs several consecutive frames to comprehend an audio segment. The Conformer (Gulati et al., 2020) has proven to be an effective local contextualization structure for audio-related tasks such as speech recognition (Gulati et al., 2020), speech translation (Fang et al., 2024), and simultaneous translation (Ma et al., 2024). Given that MoCE provides similar functionality to Conformer, it may serve as a more advantageous replacement. Moreover, tasks involving phrasal representation modeling (Huang et al., 2023b; Fang and Feng, 2022) could also benefit from MoCE's capacity as a robust multi-scale feature extractor.

7 Conclusions

In this work, we propose an Adaptive MultiScale-Headed Attention approach for effective local contextualization. It conducts contextualization on attention heads within multi-head attention. Next, we leverage the concept of MoE to achieve adaptive selection of contextualization functions. Extensive experiments demonstrate the superiority of Ada-MSHA over other byte-based models and disclose why it performs well.

Limitations

The proposed approach is only applied to encoder. How to apply local contextualization to decoder is a critical and interesting topic for future work.

References

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.

Mengyu Bu, Shuhao Gu, and Yang Feng. 2024. Improving multilingual neural machine translation by utilizing semantic and linguistic features. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 10410–10423, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Tyler A. Chang, Catherine Arnett, Zhuowen Tu, and Benjamin K. Bergen. 2023. When is multilinguality a curse? Language modeling for 250 high- and low-resource languages. *arXiv preprint*.

Jonathan H. Clark, Dan Garrette, Iulia Turc, and John Wieting. 2022. Canine: Pre-training an efficient tokenization-free encoder for language representation. *Trans. Assoc. Comput. Linguistics*, 10:73–91.

Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Y. Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loïc Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No language left behind: Scaling human-centered machine translation. *CoRR*, abs/2207.04672.

Zhenbang Du, Jiayu An, Jiahao Hong, and Dongrui Wu. 2023. Mixture-of-experts for open set domain adaptation: A dual-space detection approach. *CoRR*, abs/2311.00285.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurélien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, and et al. 2024. The llama 3 herd of models. *CoRR*, abs/2407.21783.

Lukas Edman, Gabriele Sarti, Antonio Toral, Gertjan van Noord, and Arianna Bisazza. 2024. Are character-level translations worth the wait? comparing ByT5 and mT5 for machine translation. *Transactions of the Association for Computational Linguistics*, 12:392–410.

Qingkai Fang and Yang Feng. 2022. Neural machine translation with phrase-level universal visual representations. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5687–5698, Dublin, Ireland. Association for Computational Linguistics.

Qingkai Fang, Zhengrui Ma, Yan Zhou, Min Zhang, and Yang Feng. 2024. CTC-based non-autoregressive textless speech-to-speech translation. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 9155–9161, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzmán, and Angela Fan. 2022. The Flores-101 evaluation benchmark for low-resource and multilingual machine translation. *Transactions of the Association for Computational Linguistics*, 10:522–538.

Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and Ruoming Pang. 2020. Conformer: Convolution-augmented transformer for speech recognition. In *21st Annual Conference of the International Speech Communication Association, Interspeech 2020, Virtual Event, Shanghai, China, October 25-29, 2020*, pages 5036–5040. ISCA.

Chengsong Huang, Qian Liu, Bill Yuchen Lin, Tianyu Pang, Chao Du, and Min Lin. 2023a. Lorahub: Efficient cross-task generalization via dynamic lora composition. *Preprint*, arXiv:2307.13269.

Langlin Huang and Yang Feng. 2024. Integrating multi-scale contextualized information for byte-based neural machine translation. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 9794–9801, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

Langlin Huang, Shuhao Gu, Zhang Zhuocheng, and Yang Feng. 2023b. Enhancing neural machine translation with semantic units. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2264–2277, Singapore. Association for Computational Linguistics.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *CoRR*, abs/2310.06825.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of experts. *CoRR*, abs/2401.04088.

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.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia. 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: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Jason Lee, Kyunghyun Cho, and Thomas Hofmann. 2017. Fully character-level neural machine translation without explicit segmentation. *Trans. Assoc. Comput. Linguistics*, 5:365–378.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. 2021. {GS}hard: Scaling giant models with conditional computation and automatic sharding. In *International Conference on Learning Representations*.

Shangjie Li, Xiangpeng Wei, Shaolin Zhu, Jun Xie, Baosong Yang, and Deyi Xiong. 2023. MMNMT: Modularizing multilingual neural machine translation with flexibly assembled MoE and dense blocks. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4978–4990, Singapore. Association for Computational Linguistics.

Tomasz Limisiewicz, Terra Blevins, Hila Gonen, Orevaoghene Ahia, and Luke Zettlemoyer. 2024. MYTE: Morphology-driven byte encoding for better and fairer multilingual language modeling. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15059–15076, Bangkok, Thailand. Association for Computational Linguistics.

Zhengrui Ma, Qingkai Fang, Shaolei Zhang, Shoutao Guo, Yang Feng, and Min Zhang. 2024. A non-autoregressive generation framework for end-to-end simultaneous speech-to-any translation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1557–1575, Bangkok, Thailand. Association for Computational Linguistics.

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, July 6-12, 2002, Philadelphia, PA, USA*, pages 311–318.

Sejik Park. 2024. Learning more generalized experts by merging experts in mixture-of-experts. *CoRR*, abs/2405.11530.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. 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.

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.

Prajit Ramachandran and Quoc V. Le. 2019. Diversity and depth in per-example routing models. In *International Conference on Learning Representations*.

Vikas Raunak, Siddharth Dalmia, Vivek Gupta, and Florian Metze. 2020. On long-tailed phenomena in neural machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3088–3095, Online. 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.

Phillip Rust, Jonas Pfeiffer, Ivan Vulić, Sebastian Ruder, and Iryna Gurevych. 2021. How good is your tokenizer? on the monolingual performance of multilingual language models. 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 3118–3135, Online. 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 (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational 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.

Noam Shazeer, *Azalia Mirhoseini, *Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*.

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.

Yi Tay, Vinh Q. Tran, Sebastian Ruder, Jai 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 *International Conference on Learning Representations*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

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.

Xun Wu, Shaohan Huang, Wenhui Wang, and Furu Wei. 2024. Multi-head mixture-of-experts. *ArXiv*, abs/2404.15045.

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. *Trans. Assoc. Comput. Linguistics*, 10:291–306.

Yilin Yang, Akiko Eriguchi, Alexandre Muzio, Prasad Tadepalli, Stefan Lee, and Hany Hassan. 2021. Improving multilingual translation by representation and gradient regularization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7266–7279, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Lili Yu, Daniel Simig, Colin Flaherty, Armen Aghajanyan, Luke Zettlemoyer, and Mike Lewis. 2023. MEGABYTE: predicting million-byte sequences with multiscale transformers. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

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.

Shaolei Zhang, Qingkai Fang, Zhuocheng Zhang, Zhengrui Ma, Yan Zhou, Langlin Huang, Mengyu Bu, Shangtong Gui, Yunji Chen, Xilin Chen, and Yang Feng. 2023. Bayling: Bridging cross-lingual alignment and instruction following through interactive translation for large language models. *CoRR*, abs/2306.10968.

A Language Conciseness

The Flores-100 dataset contains two subset, "dev" and "devtest". Here, we use "devtest" subset which includes 1012 parallel sentence pairs.

We first convert all sentences into byte-based strings, and then count the average sentence length for each language. Since the sentences are parallel across all languages, the average sentence lengths are comparable, showing how many bytes the language need to convey the same meaning.

Figure 3 shows a subset of languages from Flores-100. In fact, we choose the intersection set of Flores-100 and Ted-59.

B Detailed Training, Inference and Evaluation Setup

All the models and experiments are implemented based on Fairseq (Ott et al., 2019) codebase.

During training stage, we set learning rate=$5e-4$, dropout=0.1, and label smoothing=0.1. The batch size is 65536 for Ted-59 and PC-6 datasets, and 131072 for the largest OPUS-100 dataset. We apply adam optimizer with $\beta = (0.9, 0.98)$ and $\epsilon = 1e-8$. In our experiments, encoder and decoder always share the embeddings. We also apply an early-stop strategy, i.e. stop training until valid loss doesn't decrease for 10 checkpoints. Checkpoints are saved for every 5k update steps.

During inference, we average the last 5 checkpoints and evaluate with it. We apply beam search with beam size=4. We observed that byte-based NMT models tend to generate longer sequences, so we set the length penalty=1.5 for all byte-based models. Note that OPUS-100 lacks validation set and test set for these 5 languages: Aragonese(an), Divehi(dz), Yoruba(yo), Mongolian(mn), and Armenian(hy). In our experiments, these 5 languages are still trained, but we do not test them.

For comprehensive evaluation, we report BLEU (Papineni et al., 2002) and ChrF (Popović, 2015) scores using *SacreBLEU* toolkit (Post, 2018), and report COMET (Rei et al., 2022) score using *wmt22-comet-da*[5] model. To keep the article concise, we present BLEU scores in the main body, and present ChrF and COMET in the Appendix D and E, respectively.

C Applying MoCE on Subword-based Models

The experiment results in Section 5 have demonstrated the significant effectiveness of MoCE on byte-based models. In addition to that, we also explored how it benefits subword-based models. Intuitively, subword tokens contain richer information and require less contextual information. Therefore, we experimented on shorter contextual scales, i.e. $\Delta = 4$ and $\Delta = 5$. We report the BLEU scores on Table 7. According to the results, applying MoCE improves subword-based models, though the improvement is relatively minor. We conjecture this is because the subword-based vocabulary still suffers from the curse of multilinguality.

	Long	Medium	Short	L.R.	H.R.	All
Transformer	15.43	31.05	26.34	23.57	30.69	24.79
MoCE ($\Delta = 4$)	**15.71**	31.15	26.65	23.49	31.05	24.99
+*lid*	15.51	**31.46**	26.66	23.77	31.10	24.98
MoCE ($\Delta = 5$)	15.49	30.98	26.93	24.10	31.09	25.01
+*lid*	15.63	**31.46**	**27.26**	**24.26**	**31.59**	**25.34**

Table 7: BLEU scores on Ted-59 dataset. All models are subword-based.

D ChrF Scores on Ted-59 and OPUS-100

We report ChrF scores on Ted-59 and OPUS-100 test sets in Table 8 and 9, respectively.

E Comet Scores on Ted-59 and OPUS-100

We report COMET scores on Ted-59 and OPUS-100 test sets in Table 10 and 11, respectively.

[5] https://huggingface.co/Unbabel/wmt22-comet-da

	Param.	Long	Medium	Short	L.R.	H.R.	All
Transformer (Subword)	60.6M	37.44	53.05	48.15	45.74	52.41	46.70
Transformer (Byte)	44.3M	36.82	54.30	49.42	47.55	53.45	47.26
Byte-*n*CF	46.7M	35.36	53.02	48.27	46.03	52.17	46.09
MSC	44.4M	36.88	54.53	49.86	47.69	53.64	47.54
MoCE ($\Delta = 5$)	44.4M	38.47	55.07	50.60	48.19	54.27	48.30
+*lid*	44.4M	38.95	**55.14**	50.57	**48.23**	**54.45**	**48.56**
MoCE ($\Delta = 6$)	44.5M	38.98	54.80	50.54	47.78	54.29	48.34
+*lid*	44.5M	**39.04**	54.77	**50.66**	47.91	54.37	48.42

Table 8: Overall chrF scores on Ted-59 dataset. "Long", "Medium", "Short", "L.R.", "H.R." and "All" follow the same definitions as in Table 3.

	Param.	Long	Medium	Short	OPUS4	All
Transformer (Subword)	77.0M	39.97	48.02	50.36	49.13	49.96
Transformer (Byte)	44.3M	35.37	44.85	44.93	42.25	44.88
MSC	44.4M	35.39	**45.30**	45.54	43.24	45.11
MoCE ($\Delta = 5$)	44.4M	34.84	44.84	45.17	42.69	44.62
+*lid*	44.4M	35.27	45.26	45.59	42.74	45.13
MoCE ($\Delta = 6$)	44.5M	35.01	45.08	45.49	43.01	45.14
+*lid*	44.5M	**35.77**	45.27	**45.86**	**43.52**	**45.37**

Table 9: Overall chrF scores on OPUS-100 dataset. "Long", "Medium", "Short", "OPUS4" and "All" follow the same definitions as in Table 4.

	Param.	Long	Medium	Short	L.R.	H.R.	All
Transformer (Subword)	60.6M	69.41	77.64	74.49	73.58	77.39	74.46
Transformer (Byte)	44.3M	67.96	79.11	74.87	74.88	77.85	74.44
Byte-*n*CF	46.7M	66.08	77.30	73.23	72.66	76.29	72.84
MSC	44.4M	68.11	79.24	75.54	75.17	78.20	74.81
MoCE ($\Delta = 5$)	44.4M	69.74	**79.89**	76.40	75.93	79.03	75.79
+*lid*	44.4M	70.40	79.88	76.53	**76.05**	**79.34**	**76.12**
MoCE ($\Delta = 6$)	44.5M	**70.65**	79.69	76.58	75.85	79.22	76.09
+*lid*	44.5M	70.58	79.71	**76.73**	75.77	79.29	76.10

Table 10: Overall COMET scores on Ted-59 dataset. "Long", "Medium", "Short", "L.R.", "H.R." and "All" follow the same definitions as in Table 3.

	Param.	Long	Medium	Short	OPUS4	All
Transformer (Subword)	77.0M	74.42	76.22	77.53	75.86	77.00
Transformer (Byte)	44.3M	71.14	74.11	74.62	72.41	74.64
MSC	44.4M	71.21	74.42	75.34	73.12	74.69
MoCE ($\Delta = 5$)	44.4M	70.44	73.84	74.76	72.42	74.13
+*lid*	44.4M	71.06	74.41	75.18	72.88	74.82
MoCE ($\Delta = 6$)	44.5M	70.75	74.28	75.05	72.81	74.81
+*lid*	44.5M	**71.32**	**74.51**	**75.50**	**73.27**	**74.89**

Table 11: Overall COMET scores on OPUS-100 dataset. "Long", "Medium", "Short", "OPUS4" and "All" follow the same definitions as in Table 4.

F Experiment on a Small Multilingual Translation Dataset

PC-6 (Bu et al., 2024) is a small multilingual translation dataset with English and 5 other languages. The purpose of using PC-6 is mainly to investigate the effectiveness of "lid" in few-language scenario. Therefore, we only conduct byte-level preprocess, which is the same as that for Ted-59. For the same reason, we simply train and test models on solely "xx→en" direction.

Table 12 shows the experiment results on PC-6 dataset. We find that "$+lid$" does not work on this dataset. We conjecture this is because when language numbers are limited, the router already knows how to select experts and does not rely on the prior information.

	Param.	cs	kk	ro	ru	tr	All
Transformer (Byte)	44.3M	21.25	9.78	29.67	27.59	18.53	21.36
Byte-nCF	46.6M	20.65	9.30	29.01	27.19	18.36	20.91
MSC	45.0M	19.44	9.19	28.90	27.48	18.26	20.66
MoCE ($\Delta = 5$)	44.4M	**21.84**	**10.99**	**29.95**	28.22	19.17	**22.03**
+*lid*	44.4M	21.48	10.64	29.55	27.38	**19.22**	21.65
MoCE ($\Delta = 6$)	44.5M	21.09	10.66	29.76	**28.34**	18.60	21.69
+*lid*	44.5M	21.72	10.81	29.61	28.21	18.66	21.80

Table 12: Overall BLEU scores on PC-6 dataset."All" follows the same definition as in Table 3. We bold the highest BLEU scores.

G The Best Position to Place Ada-MSHA Layer

Huang and Feng (2024) has mentioned the best position to place MSC layer is the first one. Here, we conduct an experiment on Ted-59 dataset to explore the best position for Ada-MSHA layer.

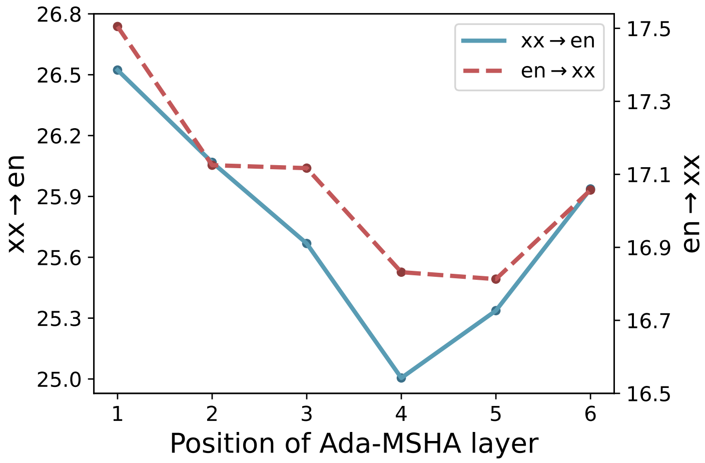


Figure 6: An empirical study on the best position to place Ada-MSHA layer. According to this figure, the first layer should be the best position, which aligns with the conclusion in MSC (Huang and Feng, 2024).

The results in Figure 6 shows that placing Ada-MSHA at the first layer is the best choice. It is not hard to figure out the explanation to this phenomenon. When placing Ada-MSHA at the first layer, the local contextualization can be treated as a part of the embedding layer, which resembles ELMo (Peters et al., 2018). At the latter layers, however, the vectors have already been globally contextualized by attention, and conducting local contextualization now makes little sense.

H Experiment Under Low Resource Setting

Intuitively, byte-based text patterns are more difficult for a model to learn, because the free combination of bytes results in a much larger language model search space. This problem should be worse under low-resource settings, where less data are utilized to constrain the model. We hypothesize that this can be mitigated by properly contextualizing, which helps models focus on local patterns. To better understand how different models perform under low-resource settings, we curated a low-resource dataset from Ted-59.

Specifically, we selected all languages whose training dataset has less than 50k parallel sentences and conducted experiments on this sub-dataset. We further split these languages into two categories: low-resource (L.R.), which has more than 10k parallel sentences, and extremely low-resource (E.L.R.), which has less than 10k parallel sentences. The corresponding languages are listed in Table 14.

The training, inference, and evaluation details are almost the same as those for Ted-59 dataset. The only difference is the representative languages for the Long, Medium, and Short categories. Since some of the languages from Table 2 are not low resources, we re-selected four languages for each category, the first two from L.R. and the last two from E.L.R., according to the order in Figure 3. For MoCE, we experiment only with the best model setting, $\Delta = 5$ with "lid".

The experiment results are exhibited in Table 13. Surprisingly, byte-based Transformer fails the tasks, while subword-based Transformer performs adequately. This demonstrates the difficulty of learning byte-based text with only global attention. Compared with byte-based Transformer, other local contextualization-based methods are able to learn the patterns with the low-resourced training data. Among them, MoCE shows significant improvement over other methods, and the advantages are larger than that on the full Ted-59 dataset. This demonstrates our proposed method, which adaptively chooses contextualization neighbors, requires fewer data to converge to an optimal solution.

	Long	Medium	Short	L.R.	E.L.R.	All
Transformer (Subword)	10.89	17.20	18.53	23.09	12.99	18.56
Transformer (Byte)	1.01	1.14	0.86	0.96	1.17	1.05
Byte-nCF	10.14	16.58	17.39	22.45	12.10	17.81
MSC	10.14	15.80	17.01	22.16	11.76	17.50
MoCE ($\Delta = 5$)	11.31	17.65	18.61	**24.03**	13.45	19.10
+*lid*	**11.80**	**18.21**	18.45	**24.03**	**13.81**	**19.45**

Table 13: The BLEU scores on the low-resource subset of Ted-59. In this table, "L.R." is the abbreviation of "Low Resource", meaning languages with training data size less than 50k but larger than 10k. "E.L.R" is the abbreviation of "Extremely Low Resource", meaning languages with training data size less than 10k. For each category, we average the scores of all languages.

Category	Languages
L.R.	et, ku, nb, sl, hy, lt, ka, hi, mk, sq, sk, gl, my, da, fi, frca
E.L.R.	bs, ur, zh, ms, bn, ta, be eu, mr, kk, eo, mn, az
Long	my, ka, ta, mr
Medium	mk, fi, kk, ms
Short	et, sl, zh, bs

Table 14: Language selection for each category.

I Effectiveness of Using Language ID

Providing Language ID (*lid*) to router seems beneficial to translation performance, according to Table 3 and Table 4. However, replacing the correct *lid* with a random one does not degenerate model performance much. Here, we take the translation direction "ro→en" as an example. The experiment is conducted on Ted-59 dataset, and we replace the *lid* for the router's input. The results are depicted in Figure 7. The green line and red dashed line denote model with correct *lid* and without *lid*, while the blue bars denote the wrong *lid* provided. It proves that having a *lid* is more critical than having the *lid* correct. We conjecture this is because the language difference is limited, and the distribution of δ is limited. However, removing the lid may cause a systematic shift of δ, similar to that discussed in Section 5.5.

In this experiment, the performance differences between different settings are nuanced, and BLEU scores are the same for several. Therefore, we evaluate with the COMET score, which is more accurate and better at showing subtle differences.

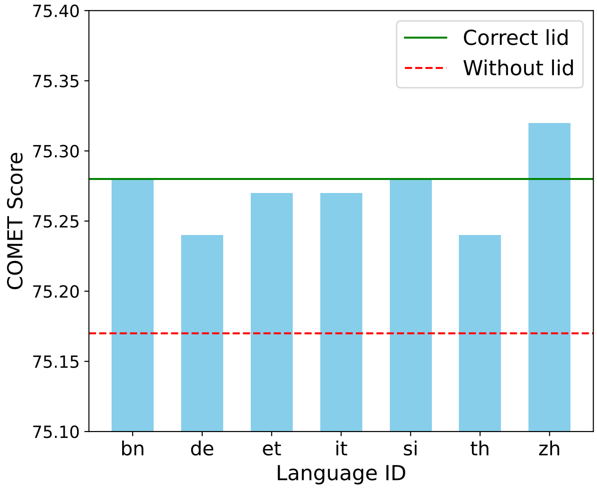


Figure 7: Experiment on "ro→en" translation. Wrong lid affects little, while lack of lid hurting a lot.