Accelerating Multilingual Language Model for Excessively Tokenized Languages

Jimin Hong **Gibbeum Lee** **Jaewoong Cho**
KRAFTON
{jimmy.h, pirensisco, jwcho}@krafton.com

Abstract

Recent advancements in large language models (LLMs) have remarkably enhanced performances on a variety of tasks in multiple languages. However, tokenizers in LLMs trained primarily on English-centric corpora often overly fragment a text into character or Unicode-level tokens in non-Roman alphabetic languages, leading to inefficient text generation. We introduce a simple yet effective framework to accelerate text generation in such languages. Our approach involves employing a new language model head with a vocabulary set tailored to a specific target language for a pre-trained LLM. This is followed by fine-tuning the new head while incorporating a verification step to ensure the model's performance is preserved. We show that this targeted fine-tuning, while freezing other model parameters, effectively reduces token fragmentation for the target language. Our extensive experiments demonstrate that the proposed framework increases the generation speed by a factor of 1.7 while maintaining the performance of pre-trained multilingual models on target monolingual tasks.

1 Introduction

Modern large language models (LLMs) (OpenAI, 2023; Touvron et al., 2023a; Antropic, 2023) have exhibited remarkable capabilities for a variety of tasks in multiple languages (Eloundou et al., 2023; Solaiman et al., 2023). Although these models are predominantly trained on English-centric data, they have shown a significant degree of multilingual proficiency (Bandarkar et al., 2023).

However, when applied to non-alphabetic languages, these models often suffer from slower text generation due to English-centric tokenization (Rust et al., 2021; Ahia et al., 2023; Petrov et al., 2023). Current tokenization techniques used in Large Language Models (LLMs) are data-driven and optimize segmentation based on the frequency of characters or bytes within a specific corpus (Sennrich et al., 2016; Kudo, 2018). As a result, the tokenizers of multilingual models, which are heavily influenced by English-dominant training data, are predominantly composed of English subwords. This leads to *excessive fragmentation*, where non-English words are overly segmented into a large number of subword units (Rust et al., 2021; Ahia et al., 2023; Petrov et al., 2023). The autoregressive nature of LLMs further amplifies this inefficiency, as it sequentially requires the generation of text.

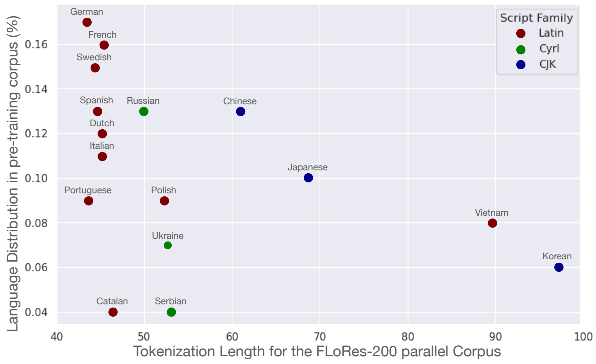

Figure 1: **Analysis of tokenization lengths and language distribution in pretraining corpus with percentage >=0.04%** English script comprises 89.7% of the corpus and has an average token length of 29.6 in FLoRes-200. The languages using the Chinese, Japanese, and Korean (CJK) scripts have longer tokenization lengths compared to those using Latin and Cyrillic scripts. Our primary focus is on languages that are excessively tokenized by English-centric tokenizers.

To address these challenges, previous studies (Wang et al., 2019; Rust et al., 2021; Cui et al., 2023) have proposed replacing or augmenting the existing vocabulary of pre-trained multilingual models with language-specific vocabularies to more effectively encode monolingual text corpora. Specifically, Rust et al. (2021) improved mBERT (Devlin et al., 2019) by replacing its tokenizer with a monolingual one and incorporating an additional 100,000 pre-training steps. On the other hand, Cui et al. (2023) enhanced Llama (Touvron

Findings of the Association for Computational Linguistics: ACL 2024, pages 11095–11111
August 11-16, 2024 ©2024 Association for Computational Linguistics

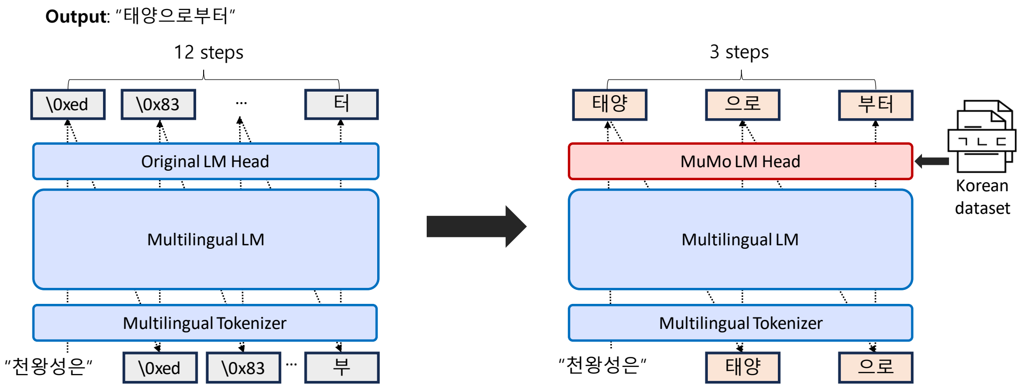

Figure 2: **Overview of the proposed framework.** Illustration of (Left) the generation with a pre-trained multilingual model and (Right) the generation of MuMo Framework. Given the Korean prefix "천왕성은" (*Uranus is*), the model generates the consecutive phrase "태양으로부터"(*from the Sun*) that consisted of 3 morphemes ("태양", "으로", "부터") in Korean. The generation with the pre-trained multilingual model faces inefficiency due to excessive fragmentation, requiring 12 steps to generate only 3 Korean morphemes. However, the MuMo framework empowers the multilingual language model to generate multiple tokens in a single iteration by extracting a word from the Korean Vocabulary, requiring 3 steps.

et al., 2023a) by expanding the Chinese vocabulary and further pre-training it on a 120GB text corpus that includes Chinese texts. However, this approach requires an extensive pre-training phase with a substantial amount of data.

Another approach to address the challenges is the use of small draft models (Leviathan et al., 2023; Chen et al., 2023a). These models generate draft output tokens, which are then verified by the original language model. However, a significant challenge arises when trying to identify or train a suitable small model that can handle multiple languages with reliable performance (Conneau et al., 2020; Bandarkar et al., 2023).

In response to these challenges, our research introduces **MuMo**, accelerating **Mu**ltilingual language models for a targeted **Mo**nolingual text generation, particularly in non-alphabetic languages. MuMo incorporates a new vocabulary of a target language into the output layer, also known as the Language Model (LM) head, and predicts the next token from this expanded vocabulary. This approach requires training only the extended portion of the output layer and specific layers of the feed-forward network. Importantly, MuMo eliminates the need for extensive text corpora or a draft model, requiring only a modest corpus of the target language, approximately 44M tokens in size. Empirical results across summarization, and translation tasks in Korean and Japanese demonstrate that MuMo significantly accelerates text generation, achieving over a 1.7-fold increase in speed without significantly compromising output quality.

Lang	Word	Multilingual Tokens
Ko	서울	("서", "\0xec", "\0xb8", "\0x9a")
Ja	発売	("発", "\0xe5", "\0xa3", "\0xb2")

Table 1: **Examples of the tokenization results.** These examples are preprocessed by the Llama tokenizer (Touvron et al., 2023b). The target monolingual word are excessively segmented into byte units, when a suitable match is not found in the multilingual vocabulary.

2 Related Work

Tokenization Disparity Subword tokenization, a common approach in LMs, is typically data-driven. Most of pre-trained tokenizers, which are often trained on predominantly English corpora, frequently result in excessive fragmentation of non-English scripts (Rust et al., 2021; Zhang et al., 2022). Ahia et al. (2023); Petrov et al. (2023) have found significant tokenization disparities across languages in popular LLMs (Xue et al., 2021, 2022; Scao et al., 2022; OpenAI, 2023). Our work endeavors to address the slowdown in inference that arises due to tokenization disparity in non-alphabetic languages.

Modifying Pre-trained Vocabulary Previous works have explored the adaptation of pre-trained vocabularies or the addition of new tokens (Artetxe et al., 2020; Rust et al., 2021; Hong et al., 2021; Liu et al., 2023), these methods often necessitate extensive pre-training to integrate the new tokens effectively (Wang et al., 2019; Chau et al., 2020; Cui et al., 2023; Liu et al., 2023). In contrast, our MuMo framework sidesteps the need for fine-tuning the parameters of pre-trained models to preserve the original capabilities of the pre-trained language model. Efforts to select items of pre-trained embedding matrix have been made (Abdaoui et al., 2020; Domhan et al., 2022; Ushio et al., 2023), but these have not yielded significant speed up where the size of the embedding layer is relatively small (Bogoychev et al., 2023).

Accelerating LLM Inference The quest to accelerate inference in auto-regressive large language models (LLMs) has led to a variety of approaches. There has been a proliferation of systems specifically engineered for LLM inference (Yu et al., 2022; Sheng et al., 2023; Xiao et al., 2023). Our proposed methodology can be harmonically integrated with the aforementioned techniques. Speculative decoding (Leviathan et al., 2023; Chen et al., 2023a) have also been explored to increase inference velocity. However, the approach often relies on the assumption that a small model can maintain high fidelity when generating a series of multiple tokens. Moreover, acquiring a small yet competitive model may be tricky, especially in a multilingual setup (Conneau et al., 2020; Bandarkar et al., 2023). Our work distinguishes itself by specifically solving the inference inefficiency that arises from excessive fragmentation in the non-alphabetic context.

Parameter Efficient Cross-lingual Transfer Learning The *curse of multilinguality*, which refers a trade-off between the language coverage and model capacity (Conneau et al., 2020), is a significant issue even in massively multilingual models, such as mBERT, XLM-R, and mT5 (Devlin et al., 2019; Conneau et al., 2020; Xue et al., 2021; Ansell et al., 2021). The problem has been mitigated through modular parameter-efficient adaptations of the multilingual models through lightweight adapters (Houlsby et al., 2019): additional trainable parameters inserted into the transformer layers of model (Pfeiffer et al., 2020; Üstün et al., 2020; Vidoni et al., 2020; Parović et al., 2022) for a target language. These techniques bear a resemblance to ours, in that they involve training partial parameters of a language model with a small amount of target language corpus. However, our goal is fundamentally different: we aim to accelerate the inference, whereas previous studies focus on improving the representational capability in target languages for multilingual models.

3 Proposed Framework

We propose a framework named **MuMo** to accelerate the inference speed of a pre-trained multilingual LM for a non-alphabetic monolingual language via a given small monolingual dataset. In the section, we introduce 1) the model architecture, 2) the fine-tuning process on a small targeted language dataset, and 3) the inference process of the proposed framework.

3.1 Model Architecture

We illustrate the model architecture of MuMo in Fig. 3.

Pre-trained Multilingual Model We consider a setting in which a pre-trained multilingual model f_{multi} is given. The model consists of 1) Transformer layers that consist of attention and feed-forward network, and 2) an output embedding layer called language model (LM) head. We denote $\mathcal{V}_{\text{multi}}$ as the multilingual vocabulary set of the model objective, as $\mathcal{L}_{\text{MLE}}(p_{\text{multi}}, \mathbf{x}) = \sum_{t=1}^{|\mathbf{x}|} \log p_{\text{multi}}(x_t|\mathbf{x}_{<t})$,

Target Monolingual LM Head The primary concept involves modifying pre-trained representations to predict a singular token unit within a target monolingual vocabulary $\mathcal{V}_{\text{mono}}$. The Target Monolingual LM head f_{mono} projects the hidden representation h, which is composed of two main components: a feed-forward network (FFN) and an output linear layer, represented as $g_{\text{mono}} : \mathbb{R}^{d_{\text{mono}}} \rightarrow \mathbb{R}^{|\mathcal{V}_{\text{mono}}|}$:

$$\text{FFN}(h) = q(W_1^\top h)W_2 \in \mathbb{R}^{d_{\text{mono}}}, \qquad (1)$$

where $W_1 \in \mathbb{R}^{d_{\text{multi}} \times d_{\text{ffn}}}$ and $W_2 \in \mathbb{R}^{d_{\text{ffn}} \times d_{\text{mono}}}$ are the weight matrices, q is non-linearity function, and d_{mono} represents the dimension of the target language representaiton. We set d_{ffn} as $d_{\text{multi}}/4$, and the non-linearity function q as SwiGLU (Shazeer, 2020). The output linear layer g_{mono} then generates a subword token:

$$f_{\text{mono}}(h) = g_{\text{mono}}(\text{FFN}(h)) \in \mathbb{R}^{|\mathcal{V}_{\text{mono}}|}. \qquad (2)$$

MuMo LM Head Note that the output space of f_{mono} is restricted to tokens in the $\mathcal{V}_{\text{mono}}$. Inspired by Lan et al. (2023), we simply extend the f_{mono} by concatenating the output linear layer of pre-trained multilingual model. This is particularly useful when there is no suitable token in $\mathcal{V}_{\text{mono}}$ to

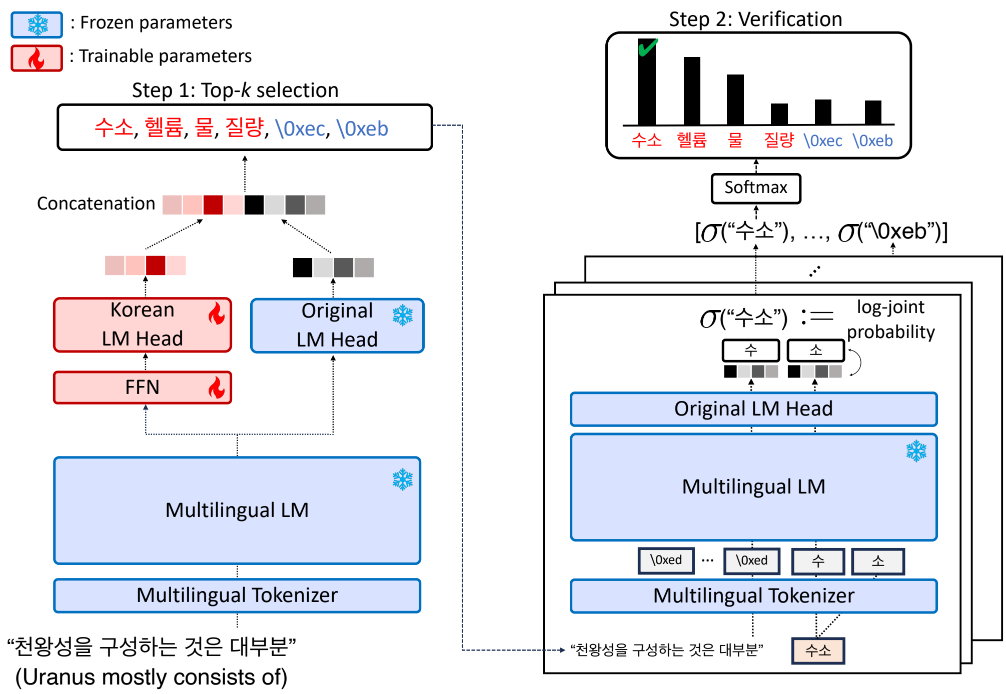

Figure 3: **Illustration of a single-step prediction with MuMo.** Initially, the MuMo LM Head f_{mumo} selects the top 6 candidates. Then, the pre-trained multilingual model verifies the feasibility of the candidates. Among the modules in MuMo, the Target Monolingual LM head (the Korean LM Head in the figure) is only trained.

predict, such as special symbols or alphabet-based tokens for non-alphabet languages.

Formally, given context representation h_{t-1}, the output of the MuMo LM head is computed as:

$$f_{\text{mumo}}(h_{t-1}) = [f_{\text{multi}}(h_{t-1}); f_{\text{mono}}(h_{t-1})] \in \mathbb{R}^{|\mathcal{V}_{\text{multi}}|+|\mathcal{V}_{\text{mono}}|} \quad (3)$$

where the symbol ; indicates the concatenation of two vectors, and the f_{mumo} indicates the output of the MuMo LM head. Thus, the MuMo LM head is composed of a combination of the pre-trained language model head and Target Monolingual LM head.

3.2 Fine-tuning

In the proposed framework, we only fine-tune the target monolingual LM head f_{mono} leveraging a small given target monolingual dataset. Note that the parameters of the pre-trained multilingual model remain frozen during the process. The model is fine-tuned by maximizing the log-likelihood of a sequence:

$$\max_{f_{\text{mono}}} \mathcal{L}_{\text{MLE}}(p_{\text{mumo}}, \mathbf{x}) = \textstyle\sum_{t=1}^{T} \log p_{\text{mumo}}(x_t|\mathbf{x}_{<t}) \;, \quad (4)$$

where $p_{\text{mumo}}(x_t|\mathbf{x}_{<t}) = \text{Softmax}(f_{\text{mumo}}(h_{t-1}))$.

3.3 Inference

Despite the availability of direct generation based on the p_{mumo}, the newly initialized Target Monolingual LM head, which is trained on limited data, may be constrained by generalization capabilities beyond the training dataset. The key concept is to leverage the probabilistic knowledge acquired by the pre-trained model p_{multi}, which has been extensively trained on large text corpora.

3.3.1 Step 1: Top-k Selection

Initially, we select top-k candidates based on the probability $p_{\text{mumo}}(x_t|\mathbf{x}_{<t})$. We set k as 10 for all experiments. Given the fact that we do not modify the input embedding of the pre-trained model, we are unable to feed the predicted word if a word does not belong in $\mathcal{V}_{\text{multi}}$ during the subsequent iteration. Instead, we input the predicted word as the tokenization units of the pre-trained vocabulary. For example, let's consider the Korean word “수소”, which corresponds to a sequence of two tokens (“수”, “소”) in $\mathcal{V}_{\text{multi}}$. If the Korean word “수소” is selected among the Top-k candidates, we employ these two multilingual tokens.

3.3.2 Step 2: Verification

Then, the *feasibility* of these potential completions is measured using the log-joint probability distribution over p_{multi}. To account for shorter sequences naturally having higher scores (Jean et al., 2015; Murray and Chiang, 2018), we normalize each candidate's score by its token length.

We measure the *feasiblity* for a candidate sequence as follows:

$$\sigma(\mathbf{c}^i) = \frac{1}{l^i} \sum_{k=1}^{l^i} \log p_{\text{multi}}(c^i_{t+k} | c^i_{<t+k}, \mathbf{x}_{<t}), \quad (5)$$

where $\mathbf{c}^i$ symbolizes a predicted token within the top-k candidates, p_{multi} represents the probability as determined by the pre-trained multilingual model, and l^i corresponds to the sequence length of the candidate $\mathbf{c}^i$.

From the k candidates, the ultimate prediction can be derived from both deterministic and stochastic manners, depending on decoding strategies.

4 Experiments

4.1 Setup

Languages As a case study, we focus on two non-roman alphabetical languages: Korean and Japanese. Since we aimed to utilize a pre-trained model with a reasonable level of effectiveness in the target language, it is essential that the language is explicitly mentioned as being trained within the pre-training corpus. In this context, we considered languages included in the Llama-2 (Touvron et al., 2023b) pre-training corpus. Moreover, the chosen language needed to exhibit the excessive fragmentation problem (Ahia et al., 2023; Petrov et al., 2023) by the English-centric pre-trained tokenizer. (See the Figure 1) This criterion led to the exclusion of most European languages such as French, German, and Portuguese. Finally, we conduct a study on multiple tasks, necessitating the existence of an instruction dataset for the target language. Due to these considerations, we only implement the experiment in Korean and Japanese.

Model We utilize the Llama-2 13B model (Touvron et al., 2023b) for all experiments. We observed some language alignment discrepancies between instructions and responses when using the Llama-2 13B chat model.[1] To address the issue, we conduct multilingual instruction tuning (Muennighoff et al., 2022) for English, Korean, and Japanese languages using the ShareGPT and Alpaca (Chen et al., 2023c). This process improve the model's fluency in each language (Muennighoff et al., 2022; Chen et al., 2023b). We also report our results test on Llama-1 13B (Touvron et al., 2023a) in Appendix.

[1]meta-llama/Llama-2-13b-chat

Language	Language Family	Pre-trained Tokenizer
Korean	Koreanic	EleutherAI/polyglot-ko-12.8b
Japanese	Japonic	rinna/japanese-gpt-neox

Table 2: **Selected languages and tokenizers.** We utilize the tokenizers to construct $\mathcal{V}_{\text{mono}}$ in each language.

Implementation of MuMo To construct targeted monolingual vocabularies in MuMo Framework, we levergage the tokenizers from the off-the-shelf model, as shown in Table 2. We selected monolingual tokens by filtering vocabulary items based on the Unicode range of each monolingual script. Additionally, we excluded items from the selection if they were already present in the pre-trained vocabulary. In terms of the preprocessing algorithm, we employ a forward maximum matching strategy to identify words in a target language vocabulary. This strategy identifies the longest sequence of tokens that aligns with a word in the target language vocabulary.

Regarding the initialization of g_{mono}, we utilize the LM head of the pre-trained multilingual model. For example, when the Korean word "태양" is tokenized into subword units ("\0xed", ..., "\0x91") using the pre-trained vocabulary, we initialize the Korean LM head of "태양" by taking the mean of the corresponding subword embeddings of the multilingual LM head. This process ensures that the initialized embeddings of Target Monolingual head represent the original word in the multilingual context.

Fine-tuning We only train the Target Monolingual LM head g_{mono} with the translated ShareGPT and Alpaca datasets (Chen et al., 2023c) in Korean, and Japanese. The training is done with 1500 steps with one batch consisting of 128 examples. We use the AdamW (Loshchilov and Hutter, 2019) optimizer with a learning rate of 0.001, weight decay of 0.01, and 150 steps of warm-up.

Evaluation We choose two representative generation tasks: summarization and translation. For summarization, we use 500 examples from XL-Sum (Hasan et al., 2021), and for translation, we use 500 examples from the FLoRes-200 (Goyal et al., 2022) dataset. We translate English sentences to each target language sentence.

Lang	Method	Summarization (0-shot)				Translation (3-shot)		
		ROUGE-2	ROUGE-L	Tokens/sec	Speed Up	BLEU	Tokens/sec	Speed Up
Ko	Vanilla Decoding	**20.7**	<u>36.1</u>	28.9	1.00x	<u>21.2</u>	29.8	1.00x
	Spec. (w/o Rejection)	18.7	33.5	<u>35.2</u>	<u>1.21x</u>	18.6	<u>36.5</u>	<u>1.22x</u>
	Spec.	20.3	35.2	27.5	0.95x	21.5	29.2	0.98x
	Shortlisting	<u>20.5</u>	**36.3**	30.6	1.06x	19.5	32.7	1.03x
	MuMo (Ours)	20.3	35.9	**55.3**	**1.92x**	**21.7**	**50.9**	**1.70x**
Ja	Vanilla Decoding	11.3	**26.6**	29.3	1.00x	**26.3**	33.4	1.00x
	Spec. (w/o Rejection)	10.8	24.2	<u>35.4</u>	<u>1.21x</u>	22.7	<u>39.9</u>	<u>1.21x</u>
	Spec.	**11.6**	<u>26.5</u>	28.5	0.97x	<u>26.0</u>	29.7	1.03x
	Shortlisting	11.4	26.3	30.3	1.03x	25.2	34.9	1.04x
	MuMo (Ours)	**11.6**	26.3	**59.2**	**2.02x**	24.3	**58.3**	**1.75x**

Table 3: **Comparative study of Language Model (LM) inference speed.** The column labeled "**Speed Up**" represents the relative performance improvement in inference speed compared to the vanilla decoding method. The highest performance in each category is highlighted in **Boldface**, and the second highest score is <u>underlined</u>. All models use sampling-based decoding. MuMo outperforms the compared baselines in the inference speed. Detailed information about the generation hyperparameters, including those used for sampling-based decoding, can be found in Appendix D.

For each task, we report 0-shot results for summarization, and 3-shot results for translation. We set the maximum sequence length as 512. We utilize flash-attention 2 (Dao, 2023) and bfloat16 types for text generation.

Metrics In the summarization task, we gauge the reliability of the generated content by calculating the **ROUGE-2** and **ROUGE-L** (Lin, 2004) scores, averaging the results across 5 different generated summaries. Likewise, for the translation task, we measure the quality of the translations by computing the **BLEU** (Papineni et al., 2002) score, again averaging over 5 translation results.[2] We report **Tokens/sec** to measure the inference speed of the models.

4.2 Baselines

We consider the following several baselines for the comparison with the proposed method. Note that all the baselines are implemented instruction-tuned model with multilingual instruction dataset (Chen et al., 2023c).

Vanilla Decoding The autoregressive generation is to sequentially sample the subsequent word based on the probability distribution over the pretrained vocabulary. This approach serves as the standard against which improvements are measured. Accounting for the nature of task, all the baselines and our framework utilizes sampling-based decoding strategy with temperature as 0.1, k as 10 for top-k sampling (Fan et al., 2018) and p as 0.7 for nucleus sampling (Holtzman et al., 2020).

Speculative Decoding Speculative decoding approach (Chen et al., 2023a; Leviathan et al., 2023) employs a preliminary "draft" model to rapidly generate a set of token candidates at each decoding step. Subsequently, these candidates undergo a validation process by the original language model to ascertain their likelihood as plausible continuations of the text. We implement two variants of this method: one with the capability to reject unsuitable candidates **(Spec.)** and another without its rejection module **(Spec. w/o Rejection)**. For the draft model, we utilize Llama-2 7B (Touvron et al., 2023b). Following the implementation of Chen et al. (2023a), we generate 5 draft tokens at each iteration.

Lexical Shortlisting Lexical Shortlisting **(Shortlisting)** (Abdaoui et al., 2020; Ushio et al., 2023), or vocabulary selection, is the approach that optimizes the decoding process by allowing it to generate a word within a set of tokens during the inference stage (Ushio et al., 2023). We implement to filter out tokens that are not present within the corresponding target language subset of the mC4 corpus (Xue et al., 2021), as Ushio et al. (2023).

4.3 Results

Table 3 shows the generation results in both summarization and translation tasks. For the summarization task in Korean, MuMo outperforms all baselines in terms of speed, achieving a 1.92x speedup over the Vanilla Decoding while maintaining competitive ROUGE scores. In translation, MuMo again demonstrates superior efficiency with a 1.70x

[2] We utilize SacreBLEU scores with the signature BLEU |nrefs:1 |case:mixed |eff:no |tok:ko,ja-mecab|smooth:exp |version:2.2.0.

Method	Update Param.	Dataset size (Tokens)	Summarization (0-shot)				Translation (3-shot)		
			ROUGE-2	ROUGE-L	Morphemes/sec	Speed Up	BLEU	Morphemes/sec	Speed Up
Vanilla Fine-tuning	13.0B	44M	**21.0**	36.0	9.8	1.00x	<u>21.4</u>	10.1	1.00x
Vocabulary Expansion	13.1B	44M	13.7	23.1	<u>17.1</u>	<u>1.92x</u>	12.3	<u>20.2</u>	<u>2.00x</u>
Vocabulary Expansion†	13.1B	60B + 44M	20.3	**37.3**	**20.5**	**2.12x**	20.3	23.1	**2.29x**
MuMo (Ours)	70M	44M	<u>20.5</u>	<u>36.3</u>	15.3	1.73x	**21.7**	17.2	1.71x

Table 4: **Comparsion with the fine-tuning strategies.** The column labeled "**Speed Up**" represents the relative performance improvement in inference speed compared to Vanilla Fine-tuning. Vocabulary Expansion† was pre-trained on over 60B tokens, comprised of both Korean and English text corpora. Other methods are only trained with the instruction dataset (44M tokens) (Chen et al., 2023c), ShareGPT and Alpaca translated in Korean. The **Boldface** signifies the superior performances, and the second highest score is <u>underlined</u>.

speed-up and even shows an improvement in BLEU score compared to Vanilla Decoding.

In the case of Japanese, the results are similar, with MuMo achieving a 2.02x speed-up in summarization and a 1.75x speed-up in translation. The ROUGE and BLEU scores for MuMo are on par with or slightly below Vanilla Decoding, indicating that the increase in speed does not significantly compromise the quality of the output.

Shortlisting shows only marginal gains in speed across both languages and every tasks, while preserving the generation capability. This is likely because the relative computational cost of processing the embedding matrix is reduced in larger models, making vocabulary reduction less impactful (Berard et al., 2021; Ushio et al., 2023). On the other hand, the **Spec.** heavily relies on the capacity of the draft model, as shown as the comparison with **(Spec. w/o Rejection)**. If the draft model lacks of sufficient multilingual capacity, it may not generate high-quality candidates, leading to a lower acceptance rate by the original model and thus reduced efficiency.

The superior performance of MuMo in terms of inference speed can be primarily attributed to its capability to predict larger linguistic units compared to those in the pre-trained vocabulary. We found that the target language tokens in $\mathcal{V}_{\text{mono}}$ are typically tokenized into 3-4 separate tokens in $\mathcal{V}_{\text{multi}}$, suggesting that the decoding step could potentially be reduced by 3-4 times. It is hypothesized that the inference speed is significantly influenced by the disparity between the pre-trained multilingual vocabulary and the target language.

5 Further Analysis

5.1 Comparative Analysis of Fine-Tuning Strategies

In the section, we provide a comparative analysis of three distinct fine-tuning strategies for multilingual models. This analysis aims to highlight the advantages and disadvantages of each strategy, particularly in terms of dataset requirements. and the number of parameters to train.

5.1.1 Setup

The two strategies compared in the analysis are:

1. **Vanilla Fine-tuning**: This strategy, which serves as a baseline, involves fine-tuning a standard multilingual model on a target monolingual instruction dataset (44M tokens) without any modifications to the pre-trained vocabulary.

2. **Vocabulary Expansion**: Inspired by prior work (Chau et al., 2020; Cui et al., 2023), this strategy involves expanding the vocabulary of the pre-trained multilingual model and fine-tuning on the instruction dataset. This method, unlike MuMo, expands not only the LM head but also the token embedding in the input layer. Two implementations of this strategy are considered. The first involves pre-training on large-scale text corpora (60B tokens)[3] before fine-tuning on the instruction dataset. This strategy is marked with a dagger in Table 4. The second only undergoes the fine-tuning phase on the instruction dataset.

To account for the variability of token unit between the different strategies, we report the inference speed with the morphemes per second **(Morphemes/sec)**, providing a standardized measurement.[4] We only compare the baselines in Korean, because of the availability of model.

5.1.2 Discussion

Table 4 reveals a consistent trend across both summarization and translation tasks. The vocabulary expansion strategies, which expand the dimension of both the token embeddings and LM head, exhibit significant increases in inference speed, but this is accompanied by a substantial decrease in the quality of the generated output when not trained on

[3]We use the off-the-shelf checkpoint from beomi/llama-2-koen-13b

[4]python-mecab-ko

	Summarization (0-shot)		Translation (3-shot)
LM Head Initialization	ROUGE-2	ROUGE-L	BLEU
Mono-init	**20.7**	36.2	21.5
Random-init	19.2	35.5	17.2
Multi-Init	20.3	**36.3**	**21.7**

Table 5: **Comparative analysis for the initialization strategy.** Mono-Init signifies to leverage the pre-existing embedding representation. We use the language model head of the monolingual model from EleutherAI/polyglot-ko-12.8b. In the case of Random-Init, we randomly initialize with Gaussian distribution. Multi-Init indicates to leverage multilingual model representation by averaging its subword embedding as the main experiment. The **Boldface** signifies the superior performances.

		Summarization (0-shot)			Translation (3-shot)	
Lang	Method	ROUGE-2	ROUGE-L	Tokens/sec	BLEU	Tokens/sec
Ko	MuMo	20.3	35.9	55.3	21.7	50.9
	w/o Verification	11.0(-9.3)	26.4(-9.5)	60.8(+5.5)	16.3(-5.4)	62.3(+11.4)
Ja	MuMo	11.6	26.3	59.2	24.3	58.3
	w/o Verification	6.7(-4.9)	20.4(-5.9)	69.1(+9.9)	10.8(-13.5)	73.6(+15.3)

Table 6: **Ablation Study**. While the exclusion of the verification accelerates approximately 1.2 times in inference speed, it significantly compromises the quality of the generation.

large-scale text corpora. This indicates that merely fine-tuning with an expanded vocabulary on a limited downstream dataset may not suffice to maintain high-quality text generation, as suggested by (Conneau et al., 2020). Furthermore, while vocabulary expansion with pre-training achieves notable speed improvements, it does not exhibit significant enhancements in generation quality.

In contrast, our proposed method exhibits a modest increase in speed while also slightly improving BLEU scores relative to vanilla fine-tuning. The principal advantage of our method lies in its capacity to attain these results without necessitating vast monolingual text corpora. This approach not only reduces the number of parameters that need to be fine-tuned, making it more parameter-efficient but also lessens the dependency on large-scale data for pre-training, making it a more data-efficient solution.

5.2 Initialization of Target Monolingual LM Head

We investigate the impact of three different initialization strategies on the target monolingual LM head g_{mono} in the Target Monolingual LM head. The first strategy involves leveraging embeddings that correspond to the pre-trained representation of a targeted monolingual LM head, termed as **Mono-init**. The second strategy is initializing the parameters with random value using Gaussian distribution (**Random-init**). Lastly, we utilize the embeddings from the pre-trained multilingual LM head (**Multi-init**), as the main experiment. This is achieved by averaging the output embeddings of the multilingual model.

Table 17 shows that Multi-init achieves a ROUGE-L score of 36.3 and a BLEU score of 21.7, which are close to the 36.2 ROUGE-L and 20.9 BLEU scores of Mono-init. On the other hand, Random-init shows a decrease in performance, with a ROUGE-L score of 35.5 and a BLEU score of 17.2.

The result demonstrates that the Multi-init approach is almost equally effective with Mono-init. This suggests that our framework can be utilized some languages that have an off-the-shelf vocabulary set but lack suitable pre-trained representations.

5.3 Effectiveness of Verification Step

We design an ablation study to investigate the role of the verification step in the inference process (Sec. 3.3.2). To assess the impact of the verification step, we generated sequences without employing the verification step.

From the results in Table 6, conducted in both Korean and Japanese, we notice that the overall generation speed is approximately 1.2 times faster when the verification is excluded. However, it is crucial to highlight that the exclusion of the verification step in the inference phase leads to a significant reduction in the generation quality. This is evident in the decrease in ROUGE-2, ROUGE-L, and BLEU scores for both languages when the verification module is not used, as shown in the table. This suggests that while the verification step may slightly slow down the generation process, it plays a vital role in preserving the model's generation capability.

5.4 Comparative Study in Single-Task Training

In the experiment, our primary objective is to investigate whether the inherent capabilities of the instruction-tuned multilingual model, which handles a variety of tasks, could be compromised when trained exclusively on single tasks using either Vocabulary Expansion or MuMo. Both methods introduce newly initialized parameters, raising concerns about potential impacts on the model's versatility. To address these concerns, we separately trained the model on each task - Question Answering (QA) (Lim et al., 2019; Kurihara et al., 2022) and Summarization (Hasan et al., 2021) - and subsequently conducted a comparative analysis between Vocabulary Expansion and MuMo.

For evaluation, we utilize multiple-task datasets, specifically Korean[5] and Japanese[6], which consist solely of questions. For the measurement, We adopt the single-answer grading setup from LLM-as-a-judge (Zheng et al., 2023). This involves presenting a question along with model-generated answers to GPT-4 (acting as the judge) for assessment. The answers are graded on a scale from 1 to 10.

As depicted in Figure 4, the instruction-tuned model initially achieves an average grading of 7.2 in the Korean experiment. However, when fine-tuned using only the QA task, Vocabulary Expansion receives a grading of 1.8, while MuMo receives a grading of 5.9. When trained solely on the summarization task, Vocabulary Expansion receives a grading of 1.6, while MuMo receives a grading of 4.7. Similar trends are observed in the Japanese experiment. The original model receives an average grading of 6.8. When fine-tuned with only the QA task, Vocabulary Expansion receives a grading of 2.1, while MuMo receives a grading of 5.2. When trained exclusively on the summarization task, Vocabulary Expansion receives a grading of 1.2, while MuMo receives a grading of 4.4.

These results suggest that while the grading of the model decreases when trained on single tasks using either method, the decrease is less pronounced with MuMo. This indicates that MuMo is more effective at preserving the model's multitask proficiency compared to Vocabulary Expansion. However, it is also clear that neither method can fully maintain the model's original instruction-following abilities on multiple tasks when trained solely on single tasks. These findings suggest that the instruction dataset, which the model was originally trained on, is crucial for preserving the pretrained model's capabilities.

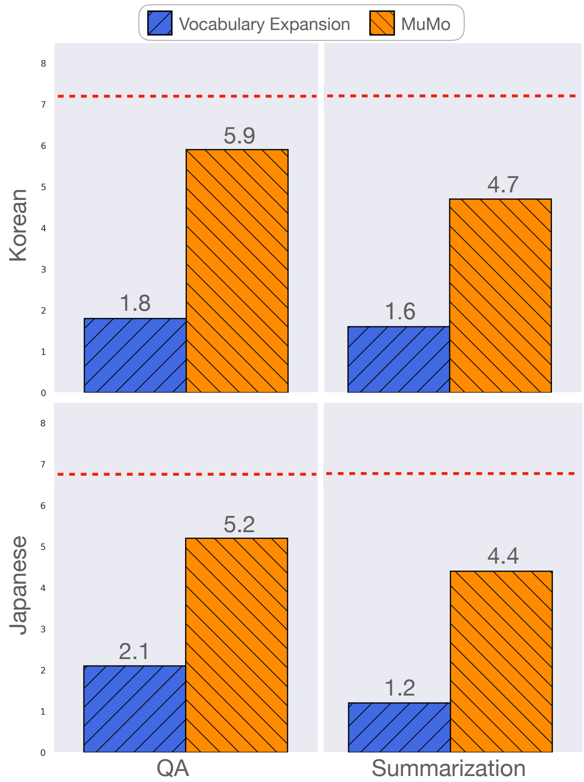

Figure 4: **Evaluation on multiple-task after training on QA and Summarization task.** The red dotted lines represent the average grading of single answers derived from the instruction-tuned multilingual language model. The decline is less pronounced with MuMo, suggesting its relative effectiveness in preserving the model's multitask proficiency.

6 Conclusion

Our study has successfully tackled the challenges in generating text for non-alphabet languages, particularly those associated with excessive fragmentation issues. The approach not only speeds up text generation but also paves the way for more efficient multilingual language applications. Our future work will broaden our experimental scope to languages that were not sufficiently represented in the pre-trained multilingual language model.

[5]Korean-MT-bench
[6]Japanese-MT-Bench

Limitations

Our proposed framework has not been evaluated with languages that exhibit excessive fragmentation issues, such as Tamil, Hebrew, and Arabic (Ahia et al., 2023; Petrov et al., 2023). These languages were not explicitly mentioned in the pre-training corpus of Llama-2 (Touvron et al., 2023b). Additionally, our framework requires off-the-shelf tokenizers for target languages to make Target monolingual LM Head. Our method does not alter the input sequence length, as we focus solely on improving the unit of prediction. This approach This approach differs from the the previous studies (Rust et al., 2021; Cui et al., 2023) which efficiently encode text at the input-level sequence length for excessively tokenized languages. Furthermore, the language models evaluated in the study are restricted to a maximum size of 13B. Larger models, such as Llama-2 30B or 70B, were not implemented due to constraints on available computational resources.

Acknowledgements

We thank Chaehun Park and Daeyoung Kim for their valuable discussions and feedback on the paper.

References

Amine Abdaoui, Camille Pradel, and Grégoire Sigel. 2020. Load what you need: Smaller versions of multilingual bert. In *Proceedings of SustaiNLP: Workshop on Simple and Efficient Natural Language Processing*.

Orevaoghene Ahia, Sachin Kumar, Hila Gonen, Jungo Kasai, David R Mortensen, Noah A Smith, and Yulia Tsvetkov. 2023. Do all languages cost the same? tokenization in the era of commercial language models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Alan Ansell, Edoardo Maria Ponti, Jonas Pfeiffer, Sebastian Ruder, Goran Glavaš, Ivan Vulić, and Anna Korhonen. 2021. MAD-G: Multilingual adapter generation for efficient cross-lingual transfer. In *Findings of the Association for Computational Linguistics: EMNLP 2021*.

Antropic. 2023. Model card and evaluations for claude models.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. On the cross-lingual transferability of monolingual representations. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Lucas Bandarkar, Davis Liang, Benjamin Muller, Mikel Artetxe, Satya Narayan Shukla, Donald Husa, Naman Goyal, Abhinandan Krishnan, Luke Zettlemoyer, and Madian Khabsa. 2023. The belebele benchmark: a parallel reading comprehension dataset in 122 language variants. *arXiv preprint arXiv:2308.16884*.

Alexandre Berard, Dain Lee, Stephane Clinchant, Kweonwoo Jung, and Vassilina Nikoulina. 2021. Efficient inference for multilingual neural machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Nikolay Bogoychev, Pinzhen Chen, Barry Haddow, and Alexandra Birch. 2023. Large language model inference with lexical shortlisting. *arXiv preprint arXiv:2311.09709*.

Ethan C. Chau, Lucy H. Lin, and Noah A. Smith. 2020. Parsing with multilingual BERT, a small corpus, and a small treebank. In *Findings of the Association for Computational Linguistics: EMNLP*.

Charlie Chen, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre, and John Jumper. 2023a. Accelerating large language model decoding with speculative sampling. *arXiv preprint arXiv:2302.01318*.

Pinzhen Chen, Shaoxiong Ji, Nikolay Bogoychev, Barry Haddow, and Kenneth Heafield. 2023b. Monolingual or multilingual instruction tuning: Which makes a better alpaca. *arXiv preprint arXiv:2309.08958*.

Zhihong Chen, Shuo Yan, Juhao Liang, Feng Jiang, Xiangbo Wu, Fei Yu, Guiming Hardy Chen, Junying Chen, Hongbo Zhang, Li Jianquan, Wan Xiang, and Benyou Wang. 2023c. MultilingualSIFT: Multilingual Supervised Instruction Fine-tuning.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Yiming Cui, Ziqing Yang, and Xin Yao. 2023. Efficient and effective text encoding for chinese llama and alpaca. *arXiv preprint arXiv:2304.08177*.

Tri Dao. 2023. Flashattention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Konstantin Dobler and Gerard de Melo. 2023. FOCUS: Effective embedding initialization for monolingual specialization of multilingual models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Tobias Domhan, Eva Hasler, Ke Tran, Sony Trenous, Bill Byrne, and Felix Hieber. 2022. The devil is in the details: On the pitfalls of vocabulary selection in neural machine translation. In *Proceedings of The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Tyna Eloundou, Sam Manning, Pamela Mishkin, and Daniel Rock. 2023. Gpts are gpts: An early look at the labor market impact potential of large language models. *arXiv preprint arXiv:2303.10130*.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation.

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*.

Tahmid Hasan, Abhik Bhattacharjee, Md Saiful Islam, Kazi Samin, Yuan-Fang Li, Yong-Bin Kang, M Sohel Rahman, and Rifat Shahriyar. 2021. Xl-sum: Large-scale multilingual abstractive summarization for 44 languages. In *Findings of the Association for Computational Linguistics, ACL)*.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Jimin Hong, Taehee Kim, Hyesu Lim, and Jaegul Choo. 2021. Avocado: Strategy for adapting vocabulary to downstream domain. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Sébastien Jean, Orhan Firat, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. 2015. Montreal neural machine translation systems for WMT'15. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Kentaro Kurihara, Daisuke Kawahara, and Tomohide Shibata. 2022. JGLUE: Japanese general language understanding evaluation. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*.

Tian Lan, Deng Cai, Yan Wang, Heyan Huang, and Xian-Ling Mao. 2023. Copy is all you need. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. 2023. Fast inference from transformers via speculative decoding. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Seungyoung Lim, Myungji Kim, and Jooyoul Lee. 2019. Korquad1. 0: Korean qa dataset for machine reading comprehension. *arXiv preprint arXiv:1909.07005*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*.

Mingjie Liu, Teo Ene, Robert Kirby, Chris Cheng, Nathaniel Pinckney, Rongjian Liang, Jonah Alben, Himyanshu Anand, Sanmitra Banerjee, Ismet Bayraktaroglu, et al. 2023. Chipnemo: Domain-adapted llms for chip design. *arXiv preprint arXiv:2311.00176*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng-Xin Yong, Hailey Schoelkopf, et al. 2022. Crosslingual generalization through multitask finetuning. *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Kenton Murray and David Chiang. 2018. Correcting length bias in neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*.

OpenAI. 2023. Gpt-4 technical report.

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*.

Marinela Parović, Goran Glavaš, Ivan Vulić, and Anna Korhonen. 2022. BAD-X: Bilingual adapters improve zero-shot cross-lingual transfer. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Aleksandar Petrov, Emanuele La Malfa, Philip HS Torr, and Adel Bibi. 2023. Language model tokenizers introduce unfairness between languages. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020. Mad-x: An adapter-based framework for multi-task cross-lingual transfer.

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 Annual Meeting of the Association for Computational Linguistics (ACL)*.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ili'c, Daniel Hesslow, Roman Castagn'e, Alexandra Sasha Luccioni, Franccois Yvon, and Matthias Gallé. 2022. Bloom: A 176b-parameter open-access multilingual language model. *ArXiv*, arXiv preprint arXiv:2211.05100.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Noam Shazeer. 2020. Glu variants improve transformer. *arXiv preprint arXiv:2002.05202*.

Ying Sheng, Lianmin Zheng, Binhang Yuan, Zhuohan Li, Max Ryabinin, Daniel Y Fu, Zhiqiang Xie, Beidi Chen, Clark Barrett, Joseph E Gonzalez, et al. 2023. High-throughput generative inference of large language models with a single gpu. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Irene Solaiman, Zeerak Talat, William Agnew, Lama Ahmad, Dylan Baker, Su Lin Blodgett, Hal Daumé III, Jesse Dodge, Ellie Evans, Sara Hooker, et al. 2023. Evaluating the social impact of generative ai systems in systems and society. *arXiv preprint arXiv:2306.05949*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Asahi Ushio, Yi Zhou, and Jose Camacho-Collados. 2023. An efficient multilingual language model compression through vocabulary trimming. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP): Findings*. Association for Computational Linguistics.

Ahmet Üstün, Arianna Bisazza, Gosse Bouma, and Gertjan van Noord. 2020. UDapter: Language adaptation for truly Universal Dependency parsing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Marko Vidoni, Ivan Vulić, and Goran Glavaš. 2020. Orthogonal language and task adapters in zero-shot cross-lingual transfer. *arXiv preprint arXiv:2012.06460*.

Hai Wang, Dian Yu, Kai Sun, Jianshu Chen, and Dong Yu. 2019. Improving pre-trained multilingual model with vocabulary expansion. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*. Association for Computational Linguistics.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2023. Efficient streaming language models with attention sinks. *arXiv preprint arXiv:2309.17453*.

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. In *Transactions of the Association for Computational Linguistics*.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mt5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Gyeong-In Yu, Joo Seong Jeong, Geon-Woo Kim, Soojeong Kim, and Byung-Gon Chun. 2022. Orca: A distributed serving system for Transformer-Based generative models. In *16th USENIX Symposium on Operating Systems Design and Implementation (OSDI 22)*.

Shiyue Zhang, Vishrav Chaudhary, Naman Goyal, James Cross, Guillaume Wenzek, Mohit Bansal, and Francisco Guzman. 2022. How robust is neural machine translation to language imbalance in multilingual tokenizer training? In *Proceedings of the 15th biennial conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS) (Datasets and Benchmarks Track)*.

Appendix

A Dataset Details

Training Data Our study employed a multilingual instruction dataset from Chen et al. (2023c), encompassing Korean and Japanese, for multilingual instruction tuning. Specifically, we utilized ShareGPT and Alpaca-GPT4 for each respective language. The dataset comprises 56k, 55k, and 168k examples for Korean, Japanese and English respectively. To train MuMo LM head, we use ShareGPT and Alpaca-GPT4 (Chen et al., 2023c) in Korean and Japanese for each language.

Evaluation Data In summarization task, we use validation and test split of XLSum (Hasan et al., 2021), which consist of 1100 examples. We found that more than half of the samples within the validation and test split surpassed the maximum sequence length of Llama-2. Consequently, we filtered out examples exceeding 1536 tokens. From the remaining examples, we randomly selected 300 for our experiments.

Regarding translation task, the dev-test set of FLoRes-200 (Goyal et al., 2022) is employed, consisting of 1012 parallel examples across both languages. We randomly use 3 examples as 3-shot prompts from training set for individual run.

When evaluating multiple-task benchmark dataset 6, we exclude examples in coding and math categories.

B Additional Results

Experiment on other Language Model Table 13, and Table 13 present the comparative study in Llama-1 13B (Touvron et al., 2023a) and Mistral 7B (Jiang et al., 2023) respectively.

Generation Results Table 15 and Table 16 present generated texts in summarization and translation tasks.

C Environment Details

All experiments are implemented using an A100-40GB GPU. The library versions utilized across all experiments include Python 3.9.10, Pytorch 2.1.0, and Transformers 4.34.0.

D Hyperparameter Details

Hyperparameter	**Value**
Learning rate	2e-5
Epoch	3
Dropout	0.1
Tensor Type	bfloat16
Batch size	128
Optimizer	AdamW
Weight decay	0.01
Warmup ratio	0.04
Maximum sequence length	2048
Learning rate scheduler	cosine

Table 7: Hyperparameters settings for multilingual instruction tuning. We follow the script from FastChat Library.

Hyperparameter	**Value**
Learning rate	1e-3
Epoch	3
Dropout	0.1
Tensor Type	bfloat16
Batch size	128
optimizer	1.05
Weight decay	AdamW
Warmup ratio	0.04
Maximum sequence length	2048
Learning rate scheduler	cosine
d_{ffn}	1280
non-linearity function q	SwiGLU

Table 8: Hyperparameters settings for training MuMo framework.

Hyperparameter	**Value**
temperature	0.1
sampling	True
p for top-p sampling	0.7
repetition penalty	1.05
exponential decay length penalty	(256, 1.03)
max sequence length	512
k for top-k sampling	20

Table 9: Hyperparameter settings for inference.

Task	Evaluation Prompt
Summarization	A chat between a curious human and an artificial intelligence assistant. The assistant gives helpful, detailed, and polite answers to the human's questions. # Document {{sourceDocument}} ## HUMAN: Summarize the document into a {{targetLang}} sentence. ## ASSISTANT:
Translation	A chat between a curious human and an artificial intelligence assistant. The assistant gives helpful, detailed, and polite answers to the human's questions. Translate the following text into {{targetLang}}. ## HUMAN: {{sourceString1}} ## ASSISTANT: {{targetString1}} ## HUMAN: {{sourceString2}} ## ASSISTANT: {{targetString2}} ## HUMAN: {{sourceString3}} ## ASSISTANT: {{targetString3}} ## HUMAN: {{sourceString}} ## ASSISTANT:

Table 10: The evaluation prompt for the main experiment (Sec. 4). We report on 0-shot results on summarization task, and 3-shot results on translation task respectively.

Task	Training Prompt
QA	A chat between a curious human and an artificial intelligence assistant. The assistant gives helpful, detailed, and polite answers to the human's questions. # Document {{context}} ## HUMAN: {{question}} ## ASSISTANT: {{answer}}
Summarization	A chat between a curious human and an artificial intelligence assistant. The assistant gives helpful, detailed, and polite answers to the human's questions. # Document {{sourceDocument}} ## HUMAN: Summarize the document into a {{targetLang}} sentence. ## ASSISTANT: {{summary}}

Table 11: The training prompt for the analysis on single-task prompt finetuning (Sec. 5.4).

		Summarization (0-shot)				Translation (3-shot)		
Lang	**Method**	**ROUGE-2**	**ROUGE-L**	**Tokens/sec**	**Speed Up**	**BLEU**	**Tokens/sec**	**Speed Up**
KO	Vanilla Decoding	14.7	31.2	29.0	1.00x	18.6	29.7	1.00x
	MuMo (Ours)	12.8	30.7	45.0	1.51x	18.1	43.0	1.49x
JA	Vanilla Decoding	10.4	21.0	28.6	1.00x	20.7	32.6	1.00x
	MuMo (Ours)	9.6	20.2	54.3	1.89x	20.0	53.8	1.64x

Table 12: **Comparative study of the inference speed in Llama-1 13B (Touvron et al., 2023a).** The column labeled "**Speed Up**" represents the relative performance improvement in inference speed compared to the vanilla decoding method.

		Summarization (0-shot)				Translation (3-shot)		
Lang	**Method**	**ROUGE-2**	**ROUGE-L**	**Tokens/sec**	**Speed Up**	**BLEU**	**Tokens/sec**	**Speed Up**
KO	Vanilla Decoding	23.0	36.5	34.2	1.00x	18.5	37.3	1.00x
	MuMo (Ours)	22.8	36.9	56.4	1.65x	18.3	63.2	1.69x
JA	Vanilla Decoding	12.9	26.4	33.5	1.00x	27.2	36.8	1.00x
	MuMo (Ours)	13.2	26.3	64.8	1.93x	26.9	66.3	1.80x

Table 13: **Comparative study of the inference speed in Mistral 7B (Jiang et al., 2023).** The column labeled "**Speed Up**" represents the relative performance improvement in inference speed compared to the vanilla decoding method.

	Texts (*ko*)	Tokens/sec	ROUGE-L
Document	환경부는 22일 사회관계장관회의에서 '1회용품 함께 줄이기 계획'을 추진한다고 발표했다. 2022년까지 일회용품 사용량을 35% 이상 줄이는 것이 정부의 목표다. 종이 일회용 컵 사용 금지 현재 카페나 빵집 등에서 일회용 플라스틱 컵은 사용이 금지되지만, 종이컵은 사용이 가능했다. 하지만 2021년부터 종이컵 제공 또한 전면 금지된다. 식당, 카페, 급식소에서 플라스틱 빨대, 젓는 막대 등도 2022년부터 금지된다. 매장에서 머그잔에 음료를 받아 마시다 포장해서 가져가려는 경우에도 일회용 컵 사용에 따른 추가 비용을 내야 한다. 환경부는 '컵 보증금제' 재도입을 검토 중이다. 소비자가 커피 등 음료를 구매할 때 일정 금액의 보증금을 내고, 컵을 반환하면 그 돈을 돌려받는 방식이다. '컵 보증금제'는 과거 한 차례 도입됐다가 2008년 폐지됐다. 포장과 음식 배달에서 제공되는 일회용 식기류 무상 제공도 2021년부터 금지된다. 정부는 배달 음식 용기 또한 친환경 소재 또는 다회용기로 전환을 유도하겠다고 발표했다. 장례식장에서도 2021년부터 일회용 식기 용품 사용이 금지된다. 비닐봉지도 금지 현재 비닐봉지는 백화점이나 슈퍼마켓 등 대규모 점포에서는 사용이 금지되어 있다. 편의점 같은 종합 소매업이나 빵집 등에서는 유상으로 구매가 가능하다. 하지만 2022년부터는 제과점이나 가게에서도 일괄 금지된다. 호텔 등 숙박업소의 경우, 50실 이상의 시설에서는 2022년부터 샴푸, 린스, 칫솔 등 일회용 위생용품 무상 제공이 금지된다. 2024년부터는 모든 숙박업소에 일괄 적용된다. 택배 포장재 줄이기 최근 택배와 신선식품 배송이 급격히 늘면서, 환경부는 배송용 포장재 사용량 증가 해소를 위한 사업도 추진한다고 발표했다. 과대포장 문제가 제기된 배송 상품의 경우 포장기준을 강화하고, 업계와 협의해 종이 완충재와 테이프 없는 상자 등 친환경 포장재를 마련할 계획이다. 2020년부터 이미 포장된 제품을 다시 포장해서 묶어 판매하는 소위 이중 포장 행위가 금지된다. 올해 13살인 라니엘은 8살 때부터 강물에 떠내려온 쓰레기를 줍기 시작했다 다른 나라는 2018년 10월, 유럽연합은 바다 오염을 막기 위해 일련의 일회용 플라스틱 제품 사용을 완전히 금하는 법안을 통과시켰다. 유럽연합은 2021년부터 법안이 발효할 것으로 기대하고 있다. 금지 품목에는 플라스틱 식기류, 빨대, 면봉 등이 있으며 식품과 음료에 사용되는 플라스틱 컵 등 일회용 플라스틱 사용 역시 줄이도록 하는 내용을 담았다. 인도의 경우 2022년부터 일회용 플라스틱 사용이 전면 금지된다.		
GT	2021년부터 카페에서 음료를 포장할 경우, 일회용 컵을 무상으로 사용하지 못한다.		
Vanila Decoding	환경부는 2022년까지 일회용품 사용량을 35% 이상 줄이는 것을 목표로 '일회용품 함께 줄이기 계획'을 추진한다고 발표했다.	27.7	33.4
MuMo	환경부에 따르면 2022년까지 일회용품 사용량에서 35% 이상 줄이기를 목표로 하며, 현재는 일회용 플라스틱 컵 사용이 금지되었으며 2021년부터는 종이컵 제공도 금지될 예정입니다.	47.2	38.1

Table 14: Generated texts on summarization task in Korean. The sample is extracted from the validation set of XLSum (Hasan et al., 2021). GT indicates the ground truth summary of the example.

	Texts (*ja*)	**Tokens/sec**	**ROUGE-L**
Document	犬のマックスは16時間 女の子に寄り添った 女の子のオーロラちゃんは前の日から行方が分からなくなり 家族や警察など約100人が捜索に当たっていた。クイーンズランド・サザンダウンズの自宅を出て原野に迷い込んだオーロラちゃんの後を 犬のマックスが追い 16時間近くずっと寄り添っていたとみられている 高齢のマックスは 目と耳が部分的に不自由。1人と1匹が丘の斜面で一緒にいるのを 親族が21日朝に発見した。オーロラちゃんの祖母 レイサ・マリー・ベネットさんは 自宅から約2キロ離れた場所で オーロラちゃんの叫び声を聞いたと豪ABCに話した。「大急ぎで山を駆け上がって上までたどりつくと 犬がこちらに向かってきて オーロラのところへ一直線に連れて行ってくれた」親類によると 気温が15度まで下がるなか オーロラちゃんは犬のマックスと岩の下に避難していたという。警察車両の横に立つマックス 動物の専門家によると 高齢な犬ほど人間とのつながりを特に重視するという クイーンズランド警察は マックスの行動を称え 名誉警察犬の地位を与えた。「3歳の子供なら 夜間とても怖かっただろうし とても寒かったはずだ 犬が寄り添っていたおかげで 女の子は心強かっただろうし 寒くならなかったのだろう 明るい結末でよかった」とクレイグ・ベリー警部は話した。ツイッターでは大勢が マックスをほめちぎり おやつをたくさんあげてほしいと書き込んだ。なぜずっとそばに なぜマックスがずっとオーロラちゃんのそばを離れなかったのかについて シドニー大学のポール・クリービー教授(動物行動学) は 高齢な犬ほど人間とのつながりを大事にするので 女の子の動揺を察知したのだろうと話す。「もし女の子が泣いていたなら 犬は元気付ける行動をとった可能性が高い」とクリービー教授はBBCに話した。「女の子のそばにずっといて 支えてあげるのが なにより大事な行動だったはずだ」		
GT	豪クイーンズランドの警察は21日午前 原野に迷い出て行方不明になった3歳少女を発見したと発表すると共に 家族の17歳になる牧牛犬が女の子に約16時間寄り添っていたと明らかにした。		
Vanila Decoding	クイーンズランド州のサザンダウンズに住む小さな女の子オーロラが 家族や警察など約100人が捜索に当たっていたが 16時間前に行方不明になった。	29.2	27.8
MuMo	オーストラリアのクイーンズランド州サバーンダウンズ地域で行方不明になった子供を見つけ オーストラリア警察の名誉警察犬になった高齢の犬マックス (Max) についてのジャーナル記事とともに オーストラリア警察の名誉警察犬になったという内容です。	57.3	31.9

Table 15: Generated texts on summarization task in Japanese. The sample is extracted from the validation set of XLSum (Hasan et al., 2021). GT indicates the ground truth summary of the example.

	Texts (*en*→*ko*)	**Tokens/sec**	**BLEU**
Source	Since moving to the Catalan-capital, Vidal had played 49 games for the club.		
GT	바르셀로나로 이적한 후 비달은 클럽을 위해 49경기를 뛰었습니다.		
Vanila Decoding	바르셀로나로 이적한 이후로 비달은 이 클럽에서 49경기에 출전했습니다.	27.2	22.8
MuMo	바르셀로나로 이적했던 비달은 클럽에서 총 49경기를 출전했습니다.	45.7	26.9

	Texts (*en*→*ko*)	**Tokens/sec**	**BLEU**
Source	Just after 11:00, protesters blocked traffic on the northbound carriage in Whitehall.		
GT	11시가 막 지난 후, 시위대는 화이트홀에 있는 북쪽으로 향하는 마차들의 교통을 막았다.		
Vanila Decoding	백알 화이트홀에서 오전 11시 15분경, 시위자들이 북쪽 차선을 차단하여 교통을 방해했습니다.	27.6	5.1
MuMo	백알 화이트홀에서 오후 11시 이후, 시위대가 북쪽 선행 차량을 차단했습니다.	44.3	4.3

	Texts (*en*→*ja*)	**Tokens/sec**	**BLEU**
Source	Since moving to the Catalan-capital, Vidal had played 49 games for the club.		
GT	カタルーニャの州都に移って以来 ビダルはクラブで49試合に出場しました。		
Vanila Decoding	バルセロナに移動してから ビダルさんは約49試合でプレーしています。	27.5	6.8
MuMo	バルセロナに移動してから ビダルさんは約49試合でプレーしています。	52.2	6.8

	Texts (*en*→*ja*)	**Tokens/sec**	**BLEU**
Source	Just after 11:00, protesters blocked traffic on the northbound carriage in Whitehall.		
GT	11時すぎちょうどに抗議者たちはホワイトホールの北行き車両の交通を遮断しました。		
Vanila Decoding	11時過ぎに ホワイトホールの北行線上で抗議者が交通を妨害しました。	28.3	26.7
MuMo	午前11時過ぎ デモ隊はホワイトホールの北へ向かう馬車の交通を阻止した。	50.3	25.0

Table 16: Generated texts on translation task. The samples are extracted from the dev-test set of FLoRes-200 (Goyal et al., 2022). GT indicates the ground truth sentence of the example.

	Summarization (0-shot)		**Translation (3-shot)**
LM Head Initialization	**ROUGE-2**	**ROUGE-L**	**BLEU**
Mono-Init	20.7	36.2	21.5
Random-Init	19.2	35.5	17.2
Multi-Init	20.3	36.3	21.7
Focus-Init	20.8	36.5	21.9

Table 17: **Comparative analysis for the initialization strategy** We exploit FOCUS (Dobler and de Melo, 2023) embedding to initialize the Target monolingual LM Head. Our framework can be harmonically integrated with the initialization strategy of multilingual token embedding.