An Empirical Study on Cross-lingual Vocabulary Adaptation for Efficient Generative LLM Inference

Anonymous ACL submission

Abstract

The development of state-of-the-art generative large language models (LLMs) disproportionately relies on English-centric tokenizers, vocabulary and pre-training data. Despite the fact that some LLMs have multilingual capabilities, recent studies have shown that their inference efficiency deteriorates when generating text in languages other than English. This results in increased inference time and costs. Cross-lingual vocabulary adaptation methods have been proposed for adapting models to a target language aiming to improve downstream performance. However, the effectiveness of these methods on increasing inference efficiency of generative LLMs has yet to be explored. In this paper, we perform an empirical study of various cross-lingual vocabulary adaptation methods on five generative LLMs (including monolingual and multilingual models) across four typologically-diverse languages and four natural language understanding tasks. We find that cross-lingual vocabulary adaptation substantially contributes to LLM inference speedups of up to 271.5%. We also show that adapting LLMs that have been pre-trained on more balanced multilingual data results in downstream performance comparable to the original models.[1]

1 Introduction

Generative large language models (LLMs) obtain strong generalization performance in many downstream natural language processing (NLP) tasks (OpenAI, 2023; Touvron et al., 2023a; Jiang et al., 2023) across various languages. For example, BLOOM (Scao et al., 2022) supports 46 languages while Open AI's ChatGPT reportedly supports 90 languages (Ahuja et al., 2023).

Despite the multilingual capabilities of state-of-the-art LLMs, their development disproportionately relies on English-oriented tokenizers, vocabulary and pre-training data. For example, around 30% of the training data in BLOOM (i.e. a multilingual LLM) is English. This negatively affects the efficiency and downstream performance of LLMs in other languages. It has been demonstrated that LLMs overfragment text in underrepresented languages with different writing systems (Rust et al., 2021; Muller et al., 2021), resulting to increased processing time, latency and costs for non-English speakers (Ahia et al., 2023; Petrov et al., 2023). Moreover, recent studies (Lin et al., 2022; Ahuja et al., 2023; Muennighoff et al., 2023) found that LLMs often perform better in a given language other than English when prompted in English instead of prompting directly in the other language. This is an unrealistic setting for non-English speakers that introduces extra disadvantages. Figure 1 shows an illustrative example of overfragmentation in non-English text generation.

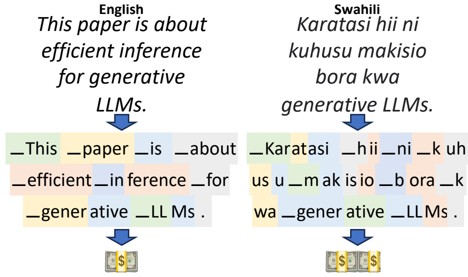

Figure 1: Example of overfragmentation when applying the Mistral-7B tokenizer to non-English text.

Cross-lingual vocabulary adaptation (Tran, 2019; Wang et al., 2020; Chau et al., 2020) is a resource-efficient method for cross-lingual transfer. The vocabulary of a source model is first updated (or replaced) with tokens from a target language, followed by fine-tuning the embedding matrix on data from the target language. Previous work on cross-lingual vocabulary adaptation methods primarily aims to improve downstream performance such as natural language inference and named-entity recog-

[1]Our code and models will be made publicly available on GitHub and Hugging Face Hub.

nition (Minixhofer et al., 2022; Dobler and de Melo, 2023). However, the effectiveness of these methods on improving inference efficiency of generative LLMs has yet to be explored. We hypothesize that LLM inference in a target language can be improved by adapting the vocabulary of the source model to reduce text overfragmentation.

To test our hypothesis, we perform an empirical study of various cross-lingual vocabulary adaptation methods, on four generative LLMs, including a wide range of downstream tasks, from text classification, and span prediction, to summarization in both zero-shot and few-shot settings across four languages (i.e. German, Japanese, Arabic, and Swahili). Our contributions are as follows:

- We demonstrate that cross-lingual vocabulary adaptation accelerates inference by up to 271.5% in 99% of cases (§5.1).
- We show that multilingual LLM vocabulary adaptation leads to comparable downstream performance to source models pre-trained on more balanced multilingual data (§5.2).
- We conduct an analysis to shed light on different design choices regarding the practical application of cross-lingual vocabulary adaptation in generative LLMs (§6).

2 Related Work

2.1 Impact of Tokenization on LLMs

Subword tokenization splits text into subword units and is the standard approach for tokenization in LLMs (Scao et al., 2022; Touvron et al., 2023a; Jiang et al., 2023). It includes methods such as WordPiece (Schuster and Nakajima, 2012), Byte Pair Encoding (BPE) (Sennrich et al., 2016), and Unigram (Kudo, 2018). Other approaches include word- (Bengio et al., 2000; Mikolov et al., 2013), character- (Al-Rfou et al., 2019) and byte-level (Xue et al., 2022) tokenization.

The impact of tokenization on LLMs has been actively studied including model performance (Bostrom and Durrett, 2020; Rust et al., 2021; Gow-Smith et al., 2022; Toraman et al., 2023; Fujii et al., 2023), inference speed (Hofmann et al., 2022; Sun et al., 2023; Petrov et al., 2023), memory usage (Sun et al., 2023),training (Ali et al., 2023) and API costs (Ahia et al., 2023; Petrov et al., 2023). It is acknowledged that tokenizers lead to disproportionate fragmentation for different languages and scripts in multi- and cross-lingual settings (Rust et al., 2021; Muller et al., 2021).

2.2 Cross-lingual Vocabulary Adaptation

Tran (2019) use English BERT as a source LM. They initialized target language token representations as a weighted sum of the source embeddings followed by fine-tuning both the source and target models. Wang et al. (2020) and Chau et al. (2020) added a fixed number of new target language tokens to the source vocabulary, expanding the source embedding matrix and output projection layers accordingly. The embeddings of the new tokens are randomly initialized over the expanded elements. Both studies performed additional pre-training on a target language corpus, often called language adaptive pre-training, i.e. LAPT (Chau et al., 2020), after the target vocabulary initialization. LAPT enables learning a target language model more efficiently than training it from scratch which is prohibitive with the size of current LLMs. It has become standard practice in more recent cross-lingual vocabulary adaptation studies (Minixhofer et al., 2022; Dobler and de Melo, 2023; Downey et al., 2023; Ostendorff and Rehm, 2023). More recently, state-of-the-art methods completely replace the source embeddings with target language embeddings instead of expanding the source vocabulary (Minixhofer et al., 2022; Dobler and de Melo, 2023; Ostendorff and Rehm, 2023; Downey et al., 2023). The aim is to utilize overlapping tokens between the source and target vocabularies for efficiency.

Cross-lingual vocabulary adaptation has been extensively used to adapt generative LLMs to specific target languages (Cui et al., 2023; Balachandran, 2023; Larcher et al., 2023). However, the majority of these approaches simply expand the source embedding matrix followed by LAPT, while vocabulary replacement approaches have not been explored. To the best of our knowledge, this is the first systematic study on the efficacy of various cross-lingual vocabulary adaptation methods for improving the inference efficiency of generative LLMs across languages.

3 Cross-lingual Vocabulary Adaptation

3.1 Problem Setting

Let $\mathcal{M}_\text{s}$ be a source pre-trained LLM with $\mathcal{T}_\text{s}$ and $\mathcal{V}_\text{s}$ its corresponding tokenizer and vocabulary. The aim is to learn a model $\mathcal{M}_\text{t}$ with the same architecture as $\mathcal{M}_\text{s}$ for a target language that supports a target vocabulary $\mathcal{V}_\text{t}$ given a tokenizer $\mathcal{T}_\text{t}$.

$\mathcal{M}_\text{t}$ is first initialized with the weights of $\mathcal{M}_\text{s}$. Subsequently, its input embedding and output layer matrices are replaced such that the former is of dimensionality $|\mathcal{V}_t| \times H_t$ and the latter $H_t \times |\mathcal{V}_t|$, where H_t is the hidden dimensionality of $\mathcal{M}_t$. The weights of both matrices are tied and they can be initialized by applying one of the target vocabulary initialization methods in §3.2. Finally, $\mathcal{M}_t$ is adapted to the target language (i.e. with LAPT) by training it on target language data $\mathcal{D}$ using a causal language modeling objective.

3.2 Target Vocabulary Initialization Methods

Random. The simplest approach is to randomly initialize the embeddings of $\mathcal{M}_t$ (de Vries and Nissim, 2021; Downey et al., 2023).

Cross-lingual and Progressive Initialization (CLP). CLP (Ostendorff and Rehm, 2023) first finds overlapping tokens between $\mathcal{V}_t$ and $\mathcal{V}_s$, i.e. $\mathcal{V}_t \cap \mathcal{V}_s$, and simply copies their weights from $\mathcal{M}_s$ to $\mathcal{M}_t$. Each target token that does not overlap with any source token, i.e. $\mathcal{V}_t \backslash (\mathcal{V}_t \cap \mathcal{V}_s)$ is initialized by its weighted average across all embeddings in $\mathcal{V}_t \cap \mathcal{V}_s$, i.e. common tokens in the source and target vocabularies. The weight of each embedding in $\mathcal{V}_t \cap \mathcal{V}_s$ is computed as the cosine similarity score between the respective overlapping token and the target non-overlapping token. Since there is no common representation between the non-overlapping token and the overlapping tokens, CLP employs vector representations from an auxiliary target language-specific pre-trained LM with the same tokenizer and vocabulary as $\mathcal{M}_t$ so that both tokens are mapped.

Heuristics. Downey et al. (2023) proposed a heuristic-based initialization that consists of the following three rules according to IDENTITY, SCRIPT, and POSITION of a token. First, embeddings are initialized according to their identity, in the same way that overlapping tokens are initialized in CLP, i.e. by copying them from $\mathcal{M}_s$. For all the remaining tokens in $\mathcal{V}_t$, their embeddings are initialized based on the type of SCRIPT identified by the Unicode block. Each token that belongs to a particular script (e.g. Hebrew) is represented by a vector sampled from a Normal distribution with the same mean and standard deviation. The mean and standard deviation of each group are computed using the embeddings of $\mathcal{M}_s$ that belong to the same group. In conjunction with SCRIPT, a group can further be divided into two according to the POSITION of each subword token in a word, i.e. whether it is placed at the beginning or in the middle (e.g. "_the" vs. "the"). Finally, the embeddings of any remaining tokens are randomly initialized.

FOCUS. Dobler and de Melo (2023) proposed fast overlapping token combinations using sparsemax (FOCUS) initialization. It is an approach similar to CLP that reuses the embeddings of $\mathcal{M}_s$ in $\mathcal{M}_t$ for tokens in $\mathcal{V}_t \cap \mathcal{V}_s$. For non-overlapping tokens $\mathcal{V}_t \backslash (\mathcal{V}_t \cap \mathcal{V}_s)$, FOCUS uses fastText (Bojanowski et al., 2017) trained on target specific data $\mathcal{D}$ tokenized by $\mathcal{T}_t$ and computes the cosine similarities between tokens in $\mathcal{V}_t \cap \mathcal{V}_s$ and $\mathcal{V}_t \backslash (\mathcal{V}_t \cap \mathcal{V}_s)$ in the fastText model. It then applies sparsemax (Martins and Astudillo, 2016), which is a sparse variant of softmax that assigns zero to any low-probability elements, over the similarity scores. The embeddings of tokens in $\mathcal{V}_t \backslash (\mathcal{V}_t \cap \mathcal{V}_s)$ are finally initialized by taking the weighted sum of the source embeddings of tokens in $\mathcal{V}_t \cap \mathcal{V}_s$, whose weights are the similarity scores with sparsemax applied.

CLP+. Finally, we propose *CLP+*, a modification to CLP motivated by the use of sparsemax in FOCUS. The aim is to dynamically select semantically similar tokens from $\mathcal{V}_t \cap \mathcal{V}_s$ to initialize a target embedding for a token in $\mathcal{V}_t \backslash (\mathcal{V}_t \cap \mathcal{V}_s)$, leading to a better initialization of the embeddings (Tran, 2019). We follow the same process as CLP for tokens in $\mathcal{V}_t \cap \mathcal{V}_s$. For non-overlapping tokens in $\mathcal{V}_t \backslash (\mathcal{V}_t \cap \mathcal{V}_s)$, instead of taking the weighted average of *all* overlapping source embeddings of $\mathcal{V}_t \cap \mathcal{V}_s$ as in CLP, we use the weighted sum of embeddings whose weight is calculated with sparsemax.

4 Experimental Setup

4.1 Source Models

We first use **BLOOM-1B** and **BLOOM-7B** (Scao et al., 2022) as source models, which are trained on data from 46 languages including Arabic (4.6%) and Swahili (0.02%). We also use **TigerBot-7B** (Chen et al., 2023), which is based on LLaMA 2 (Touvron et al., 2023b) adapted using data from East Asian languages, i.e. Chinese (54%), Korean (0.001%), and Japanese (0.01%). Finally, we experiment with **Mistral-7B** (Jiang et al., 2023) which is an English-centric model. Table 1 shows the tokenizer and vocabulary size of each source model.

4.2 Target Languages and Adaptation Data

We experiment with a typologically diverse set of target languages including German (Indo-European), Japanese (Japonic), Arabic

Source ($\mathcal{M}_s$)	Tokenizer ($\mathcal{T}_s$)	$\lvert\mathcal{V}_s\rvert$
BLOOM	Byte-level BPE	250,680
TigerBot	Byte-level BPE	60,512
Mistral	Byte-level BPE	32,000
Target ($\mathcal{M}_t$)	**Tokenizer ($\mathcal{T}_t$)**	$\lvert\mathcal{V}_t\rvert$
German	Byte-level BPE	50,257
Japanese	Unigram	32,000
Arabic	Byte-level BPE	64,000
Swahili	Byte-level BPE	50,257

Table 1: Tokenizers and vocabulary size for source and target models.

(Afro–Asiatic), and Swahili (Niger–Congo). We opted to use these languages because of the availability of language-specific (1) tokenizers; and (2) downstream task datasets with the same task formulation across languages.[2]

For adapting the source models, we use the OSCAR language-specific subcorpus (Jansen et al., 2022) for German, Arabic, and Japanese (January 2023 version). For Swahili, we use the Swahili subset of CC-100 (Conneau et al., 2020) following Minixhofer et al. (2022). We use publicly available existing tokenizers and vocabularies for each target language. Table 1 shows the type of tokenizer and vocabulary size for the target models. More details are available in Table 3 in the Appendix.

4.3 Tasks

Following Ahia et al. (2023), we evaluate all models including baselines across four tasks in each target language: (1) textual entailment (NLI) consisting of JNLI (Kurihara et al., 2022) for Japanese and XNLI (Conneau et al., 2018) for the rest; (2) X-CSQA (Lin et al., 2021) for multiple choice question-answering (MC); (3) summarization (SUM) including MLSUM (Scialom et al., 2020) for German and XL-Sum (Hasan et al., 2021) for the rest; and (4) span prediction (SPAN) consisting of XQuAD (Kurihara et al., 2022) for Arabic and German, JSQuAD (Kurihara et al., 2022) for Japanese and KenSwQuAD (Wanjawa et al., 2023) for Swahili. Similarly, we use 500 random samples.

Due to the computational constraints, we conduct zero-shot experiments on SUM. For the rest of the tasks, we evaluate models in zero- and few-shot settings. We use five demonstrations for NLI and MC and three for SPAN in the few-shot cases.

[2]Note that data for the same task across languages does not match. Model performance is not directly comparable.

4.4 Prompt Templates

We use the same English prompt templates as Ahia et al. (2023) for NLI and SUM. For MC and SPAN, we formulate a task-specific English prompt. We translate the English prompt templates into each corresponding target language using a machine translation API (i.e. Google Translate), following Yong et al. (2023). The prompt templates can be found in Appendix A.7.

4.5 Baselines

We compare the cross-lingual vocabulary adaptation methods against two baselines: (1) we use the source models directly on the target language tasks without any adaptation (**Source**); and (2) following Yong et al. (2023), we adapt the source models by continuing pre-training on data from a target language by keeping the source vocabulary (**LAPT**).

4.6 Evaluation Metrics

Inference Efficiency. We calculate the average number of prompt tokens per sample for each dataset and tokenizer, and use its relative ratio to each source tokenizer as a proxy for inference speedup following (Ahia et al., 2023; Petrov et al., 2023). We use the average number of prompt tokens rather than the actual inference time because commercial APIs (e.g. OpenAI) often charge users on the basis of the total number of prompt and generated tokens. Note that inference efficiency is independent of the model size.

Downstream Performance. For downstream performance evaluation, we use standard metrics for each dataset such as accuracy for NLI and MC, F1 for SPAN, and ROUGE-L for SUM.

4.7 Implementation Details

Efficient LLM Adaptation. We perform our experiments under resource-constrained settings due to limited access to computational resources. For computational efficiency, we use a low-rank adaptation approach with LAPT, i.e. LoRA (Hu et al., 2021) applied on all linear layers (setting rank $r = 8$), following (Yong et al., 2023; Cui et al., 2023; Balachandran, 2023; Larcher et al., 2023). We pre-train each model for a maximum of four days. We use a batch size of 8 for BLOOM-1B and 16 for the 7B models with gradient accumulation steps set to 4 and a sequence length of 1,024. We set the learning rate to 1e-4 and save checkpoints every 1,000 steps. For a fair comparison,

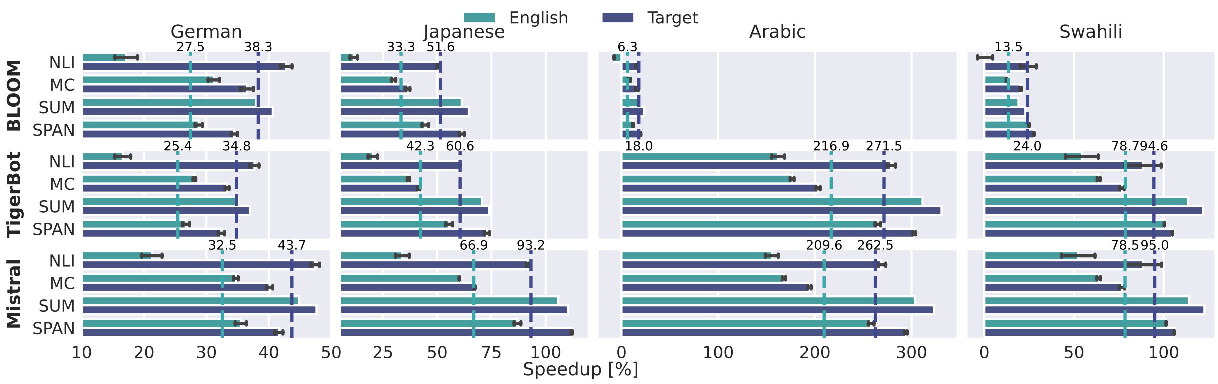

Figure 2: Relative speedup ratios to each base model/tokenizer when prompted in English and a target language. Dotted lines denote the average speedup ratio across tasks in each setting.

we use the checkpoints with the largest number of steps available across all vocabulary initialization approaches and the LAPT baseline for the same source model size and language.[3]

Libraries and Hardware. We implement our models using PyTorch (Paszke et al., 2019), HF Transformers (Wolf et al., 2020) and PEFT (Mangrulkar et al., 2022). We preprocess data with HF Datasets (Lhoest et al., 2021). We use a single NVIDIA A100 (80GB) GPU for all experiments.

5 Results

5.1 Inference Efficiency

Figure 2 shows the relative inference speedup ratio between the target and source model prompted in the target language and English. Overall, the results confirm our hypothesis that cross-lingual vocabulary adaptation accelerates inference in 95 out of 96 cases including zero- and few-shot settings.

Next, we examine the efficiency of target models in each language. We first observe that the models adapted in German show moderate average speedup ratios (25.4-43.7%) across different tasks, source models and prompting languages. This is possibly due to the close relationship between German and English (i.e. both Germanic and Indo-European languages). The Japanese target models also exhibit moderate but slightly greater average speedups compared to German of up to 60.6% using BLOOM and TigerBot as source models. In contrast, inference speedups are substantially greater using Mistral as the source model (66.9-93.2% on average). These differences can stem from the inclusion of Chinese pre-training data in BLOOM,[4] and Chinese and Japanese data in TigerBot. Target models in Arabic and Swahili obtain smaller speedups than the other languages (i.e. up to 24.0% on average) using BLOOM as the source model. This is likely due to the inclusion of Arabic and Swahili pre-training data in BLOOM. In contrast, target models in both languages obtain substantial gains using TigerBot and Mistral as the source models, i.e. up to an impressive 271.5% for Arabic and 95.0% for Swahili. We speculate that this happens because the two languages are not included in the training data of TigerBot and Mistral[5], and the different Arabic script.

Looking into individual tasks, we observe that adapted models gain larger speedup ratios in SPAN and SUM compared to the other two tasks across source models and languages. In particular, we record a maximum speedup of 331% in Arabic SUM with in-language prompting using TigerBot as the source model. In contrast, speedup ratios tend to be smaller than average in NLI and MC across different target and source models, except for NLI with in-language prompting. Specifically, the Arabic model using BLOOM as source shows a slowdown of 7.63% when prompting in English. Our hypothesis is that this is due to the ratio of English-related words included in a prompt in each task, resulting to overfragmentation of such words by a *target-language* tokenizer, which is detrimental to inference speedup. Indeed, the number of tokens of the NLI English prompt template[6] is ten when tokenized with the BLOOM source tokenizer,

[3] For more details, refer to Appendix A.4.

[4] Note that the Japanese script includes Chinese characters.

[5] We do not have enough information about the training data of the model. Mistral.ai states that it is good at English tasks according to the blog post.

[6] `Question: True, False, or Neither? Answer:`

Approach	German NLI	German MC	German SUM	German SPAN	Japanese NLI	Japanese MC	Japanese SUM	Japanese SPAN	Arabic NLI	Arabic MC	Arabic SUM	Arabic SPAN	Swahili NLI	Swahili MC	Swahili SUM	Swahili SPAN
BLOOM-1B								**Zero-shot**								
Source	.33	.21	17.8	.06	.29	.20	18.2	.22	.35	.20	12.0	.15	.32	.22	12.0	.03
LAPT	.36	.22	14.3	.09	.28	.20	20.7	.26	.36	.19	11.4	.13	.31	.18	7.7	.07
+ Random	.35$_{6.1}$	**.22**$_{4.8}$	15.3$_{15.0}$	**.14**$_{133.3}$	**.29**$_{0.0}$	**.21**$_{5.0}$	19.0$_{5.6}$	**.32**$_{45.5}$	.35$_{0.0}$	.19$_{5.0}$	11.5$_{4.2}$	.14$_{6.7}$	**.33**$_{3.1}$	**.22**$_{0.0}$	10.2$_{15.0}$	**.08**$_{166.7}$
+ Heuristics	.34$_{3.0}$	.19$_{9.5}$	15.3$_{15.0}$	**.13**$_{116.7}$	**.29**$_{0.0}$	.19$_{5.0}$	19.2$_{6.7}$	**.31**$_{40.9}$	**.36**$_{2.9}$	**.22**$_{10.0}$	11.3$_{5.8}$	.13$_{13.3}$	**.34**$_{6.3}$	**.22**$_{0.0}$	11.9$_{0.8}$	**.11**$_{266.7}$
+ CLP	.34$_{3.0}$	.18$_{14.3}$	14.6$_{18.9}$	**.14**$_{133.3}$	**.29**$_{0.0}$	**.25**$_{25.0}$	18.8$_{4.4}$	**.33**$_{50.0}$	**.36**$_{2.9}$	**.21**$_{5.0}$	11.2$_{6.7}$	.14$_{6.7}$	**.34**$_{6.3}$	**.22**$_{0.0}$	11.5$_{4.2}$	**.11**$_{266.7}$
+ FOCUS	.34$_{3.0}$	.19$_{9.5}$	16.1$_{10.6}$	**.13**$_{116.7}$	**.29**$_{0.0}$	**.21**$_{5.0}$	19.2$_{6.7}$	**.33**$_{50.0}$	.35$_{0.0}$	**.20**$_{0.0}$	11.2$_{6.7}$	.14$_{6.7}$	**.34**$_{6.3}$	**.22**$_{0.0}$	11.2$_{6.7}$	**.12**$_{300.0}$
+ CLP+	.31$_{6.1}$	.15$_{28.6}$	15.8$_{12.2}$	**.13**$_{116.7}$	**.29**$_{0.0}$	.19$_{5.0}$	19.4$_{7.8}$	**.33**$_{50.0}$	.35$_{0.0}$	.17$_{15.0}$	11.3$_{5.8}$	**.15**$_{0.0}$	**.34**$_{6.3}$	.20$_{9.1}$	10.4$_{13.3}$	**.10**$_{233.3}$
BLOOM-7B																
Source	.36	.21	23.1	.15	.28	.21	19.0	.33	.38	.17	11.5	.25	.36	.22	14.3	.22
LAPT	.37	.21	19.4	.14	.21	.21	21.6	.36	.36	.16	11.5	.21	.36	.20	13.0	.14
+ Heuristics	.32$_{11.1}$	**.22**$_{4.8}$	19.7$_{14.3}$	**.21**$_{40.0}$	**.30**$_{7.1}$	**.23**$_{9.5}$	19.5$_{2.6}$	**.38**$_{15.2}$	.37$_{2.6}$	**.19**$_{11.8}$	10.7$_{2.7}$	.21$_{16.0}$	.33$_{8.3}$	**.22**$_{0.0}$	11.6$_{17.1}$	.16$_{27.3}$
+ CLP+	.35$_{2.8}$	**.21**$_{0.0}$	18.7$_{18.7}$	**.20**$_{33.3}$	**.29**$_{3.6}$	**.21**$_{0.0}$	19.5$_{2.6}$	**.40**$_{21.2}$	**.38**$_{0.0}$	**.21**$_{23.5}$	11.0$_{0.0}$	.21$_{16.0}$	.33$_{8.3}$	**.23**$_{4.5}$	10.9$_{22.1}$	.17$_{22.7}$
TigerBot-7B																
Source	.33	.24	23.9	.26	.17	.24	19.4	.57	.33	.21	9.0	.04	.29	.22	12.4	.03
LAPT	.32	.21	18.5	.18	.17	.21	21.6	.49	.33	.18	9.8	.13	.31	.21	15.9	.10
+ Heuristics	**.35**$_{6.1}$	.20$_{16.7}$	16.1$_{32.9}$	.18$_{30.8}$	**.29**$_{70.6}$	.22$_{8.3}$	19.6$_{3.2}$	.40$_{29.8}$	**.36**$_{9.1}$	.17$_{19.0}$	**10.3**$_{14.4}$	.08$_{100.0}$	**.36**$_{24.1}$	**.22**$_{0.0}$	8.1$_{32.5}$	.05$_{66.7}$
+ CLP+	**.33**$_{0.0}$	.20$_{16.7}$	14.1$_{41.2}$	.19$_{26.9}$	**.29**$_{70.6}$	.20$_{16.7}$	19.8$_{4.2}$	.41$_{28.1}$	**.38**$_{15.2}$	**.22**$_{4.8}$	**11.2**$_{24.4}$	**.16**$_{300.0}$	.30$_{3.4}$	**.22**$_{0.0}$	8.6$_{28.3}$	.09$_{200.0}$
Mistral-7B																
Source	.34	.25	24.1	.35	.17	.28	23.7	.60	.33	.20	11.2	.21	.35	.22	15.4	.07
LAPT	.33	.25	24.2	.28	.17	.20	23.4	.60	.33	.18	10.8	.14	.33	.22	16.2	.12
+ Heuristics	**.40**$_{17.6}$	**.26**$_{4.0}$	21.2$_{11.7}$	.22$_{37.1}$	**.29**$_{70.6}$	.20$_{28.6}$	19.7$_{17.9}$	.43$_{28.3}$	**.39**$_{18.2}$	.19$_{5.0}$	10.7$_{2.7}$	.13$_{38.1}$	.34$_{2.9}$	**.22**$_{0.0}$	10.6$_{29.3}$	**.14**$_{100.0}$
+ CLP+	**.35**$_{2.9}$	**.25**$_{0.0}$	20.2$_{15.8}$	.21$_{40.0}$	**.28**$_{64.7}$	.20$_{28.6}$	19.9$_{17.1}$	.46$_{23.3}$	**.38**$_{15.2}$	.16$_{20.0}$	**11.5**$_{4.5}$	**.21**$_{0.0}$	.33$_{5.7}$	.21$_{4.5}$	10.2$_{32.0}$	**.16**$_{128.6}$
BLOOM-1B								**Few-shot**								
Source	.38	.20	-	.10	.44	.19	-	.32	.35	.17	-	.20	.37	.23	-	.02
LAPT	.37	.17	-	.13	.26	.21	-	.34	.34	.16	-	.16	.34	.19	-	.02
+ Random	.34$_{10.5}$	**.21**$_{5.0}$	-	**.16**$_{60.0}$	.29$_{34.1}$	**.21**$_{10.5}$	-	**.34**$_{6.3}$	**.35**$_{0.0}$	**.22**$_{29.4}$	-	.16$_{20.0}$	.34$_{8.1}$	.20$_{13.0}$	-	**.06**$_{200.0}$
+ Heuristics	.33$_{13.2}$	**.23**$_{15.0}$	-	**.17**$_{70.0}$	.30$_{31.8}$	**.22**$_{15.8}$	-	.32$_{0.0}$	**.36**$_{2.9}$	**.21**$_{23.5}$	-	.15$_{25.0}$	.33$_{10.8}$	.19$_{17.4}$	-	**.07**$_{250.0}$
+ CLP	.34$_{10.5}$	**.21**$_{5.0}$	-	**.17**$_{70.0}$	.30$_{31.8}$	.20$_{5.3}$	-	.33$_{3.1}$	**.35**$_{0.0}$	**.21**$_{23.5}$	-	.15$_{25.0}$	.33$_{10.8}$	.19$_{17.4}$	-	**.08**$_{300.0}$
+ FOCUS	.34$_{10.5}$	.18$_{10.0}$	-	**.17**$_{70.0}$	.27$_{38.6}$	.20$_{5.3}$	-	**.36**$_{12.5}$	**.36**$_{2.9}$	**.20**$_{17.6}$	-	.15$_{25.0}$	.34$_{8.1}$	.19$_{17.4}$	-	**.08**$_{300.0}$
+ CLP+	.34$_{10.5}$	**.20**$_{0.0}$	-	**.19**$_{90.0}$	.29$_{34.1}$	**.22**$_{15.8}$	-	**.36**$_{12.5}$	**.37**$_{5.7}$	**.20**$_{17.6}$	-	.15$_{25.0}$	.30$_{18.9}$	.18$_{21.7}$	-	**.08**$_{300.0}$
BLOOM-7B																
Source	.38	.23	-	.29	.41	.19	-	.49	.37	.18	-	.29	.34	.18	-	.11
LAPT	.35	.24	-	.23	.34	.19	-	.53	.36	.18	-	.23	.37	.18	-	.07
+ Heuristics	.33$_{13.2}$	.22$_{4.3}$	-	.28$_{3.4}$	.28$_{31.7}$	**.21**$_{10.5}$	-	.46$_{6.1}$	.36$_{2.7}$	**.21**$_{16.7}$	-	.24$_{17.2}$	.36$_{5.9}$	**.19**$_{5.6}$	-	**.13**$_{18.2}$
+ CLP+	.34$_{10.5}$	.22$_{4.3}$	-	.25$_{13.8}$	.30$_{26.8}$	**.20**$_{5.3}$	-	.46$_{6.1}$	.36$_{2.7}$	**.22**$_{22.2}$	-	.25$_{13.8}$	.36$_{5.9}$	**.18**$_{0.0}$	-	**.13**$_{18.2}$
TigerBot-7B																
Source	.31	.37	-	.42	.16	.34	-	.65	.30	.19	-	.10	.36	.19	-	.03
LAPT	.30	.39	-	.36	.16	.34	-	.66	.30	.20	-	.17	.36	.20	-	.09
+ Heuristics	**.33**$_{6.5}$	.26$_{29.7}$	-	.21$_{50.0}$	**.29**$_{81.2}$	.24$_{29.4}$	-	.49$_{24.6}$	**.35**$_{16.7}$	**.20**$_{5.3}$	-	.09$_{10.0}$	.35$_{2.8}$	**.21**$_{10.5}$	-	.04$_{33.3}$
+ CLP+	**.36**$_{16.1}$	.31$_{16.2}$	-	.31$_{26.2}$	**.30**$_{87.5}$	.21$_{38.2}$	-	.50$_{23.1}$	**.37**$_{23.3}$	.19$_{0.0}$	-	**.19**$_{90.0}$	.34$_{5.6}$	.18$_{5.3}$	-	.06$_{100.0}$
Mistral-7B																
Source	.33	.53	-	.48	.16	.42	-	.69	.30	.32	-	.31	.40	.21	-	.12
LAPT	.33	.46	-	.27	.16	.37	-	.68	.30	.30	-	.26	.36	.34	-	.21
+ Heuristics	**.45**$_{36.4}$	.41$_{22.6}$	-	.24$_{50.0}$	**.30**$_{87.5}$	.24$_{42.9}$	-	.49$_{29.0}$	**.34**$_{13.3}$	.18$_{43.8}$	-	.17$_{45.2}$	.34$_{15.0}$	.18$_{14.3}$	-	.09$_{25.0}$
+ CLP+	**.37**$_{12.1}$	.47$_{11.3}$	-	.25$_{47.9}$	**.29**$_{81.2}$	.25$_{40.5}$	-	.50$_{27.5}$	**.38**$_{26.7}$	.23$_{28.1}$	-	.23$_{25.8}$	.33$_{17.5}$	.20$_{4.8}$	-	.14$_{16.7}$

Table 2: Mean performance over five runs with in-language prompting on 500 randomly selected test samples for each dataset. The baselines are in grey. **Bold** indicates comparable or better results than the baselines. Green indicates positive relative performance change over Source. Red denotes negative relative performance change.

increasing to 21 with the Arabic target tokenizer.

Finally, we investigate the inference efficiency for the target models by prompting language. We observe that the target models show greater inference speedup ratios with in-language prompts than English in all cases. The average differences between in-language and English prompts are 12.8%, 24.6%, and 26.8%, using BLOOM, TigerBot, and Mistral as source models, respectively. This suggests that the target models are susceptible to code-mixed text (i.e. including English prompts), leading to overfragmentation for words not in a target language. Furthermore, in-language prompting is a more realistic scenario for non-English speakers to use LLMs than English prompting. Therefore, these differences reflect the advantage of cross-lingual vocabulary adaptation and confirm the disparity of using a source tokenizer, reported by Ahia et al. (2023) and Petrov et al. (2023).

5.2 Downstream Performance

We compare the downstream performance of all cross-lingual vocabulary adaptation methods (§3.2) to the baselines, i.e. Source and LAPT (§4.5) using BLOOM-1B as the source model. Due to computational costs, we only evaluate the best two vocabulary adaptation approaches (Heuristics and CLP+) against the baselines with larger source models (BLOOM-7B, TigerBot-7B and Mistral-7B). Table 2 shows the zero- and few-shot performance of all models with in-language prompting. Results using English prompts are included in the Appendix (Table 9). We examine differences between English and target language prompting in §6.

Overall, adapted models show comparable or better performance than the baselines in the majority of the cases across tasks and languages using BLOOM-1B as source. Models adapted with simple Random target vocabulary initialization are competitive compared to more sophisticated approaches and the baselines in the majority of the cases – 18 for Source and 24 for LAPT out of 28 cases, respectively. However, they are not as robust with English prompting (see Table 9 in the Appendix). Models adapted with Heuristics also perform competitively with the semantic similarity-based methods (i.e. CLP, FOCUS and CLP+). They are similar to or better than Source in 17 out of 28 cases and LAPT in 19 out of 28 cases without a substantial drop in performance observed in Ran-

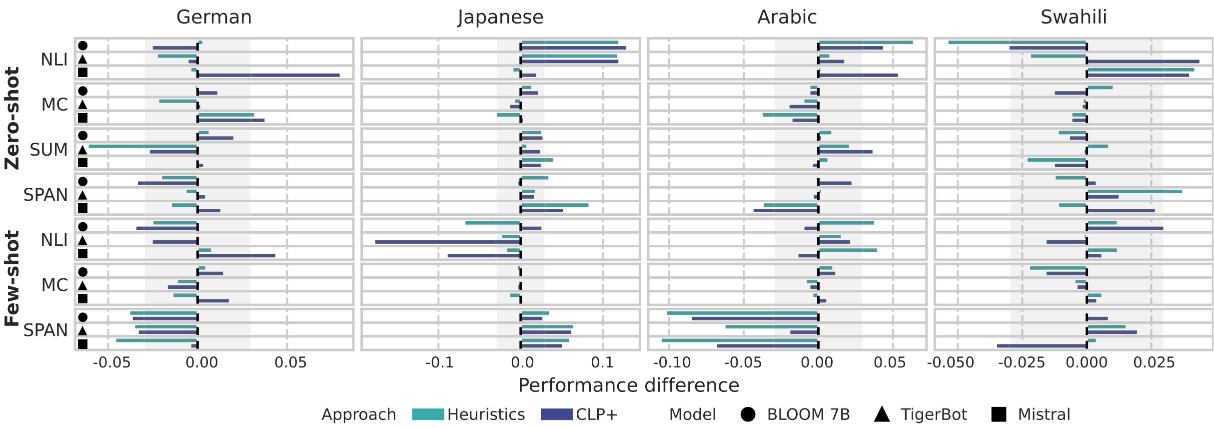

Figure 3: Performance difference between English and in-language prompts. Positive and negative values indicate better performance using in-language or English prompts respectively.

dom with English prompting. These results generally corroborate findings by Downey et al. (2023) that (1) Random initialization is not robust, and (2) Heuristics rivals the semantic similarity-based methods. CLP+ outperforms CLP and FOCUS in the few-shot cases excluding Random, while Heuristics is the best zero-shot approach.

Experimenting with larger source models, we observe that models adapted with Heuristics perform competitively with CLP+ with BLOOM-7B. They also show competitive performance with LAPT in 18 out of 28 cases across tasks and languages. However, they exhibit slightly worse performance than Source in 15 out of 28 cases. Models adapted with CLP+, in contrast, perform on par with Source (14 out of 28 cases) and slightly better than LAPT (17 out of 28 cases). When using TigerBot and Mistral as source, we find that adapted models fail to perform better than the baselines. The only exception is TigerBot, where models adapted with Heuristics and CLP+ show competitive performance with Source in 15 and 16 out of 28 cases across tasks and languages. This suggests that LLMs not as multilingual as BLOOM (i.e. TigerBot and Mistral) may not perform well with cross-lingual vocabulary adaptation, possibly due to less transferable cross-lingual knowledge, and the small amount of target language data included during pre-training.

6 Analysis

In-language vs. English Prompting. Recent studies (Lin et al., 2022; Ahuja et al., 2023; Muennighoff et al., 2023) report that in-language prompting yields lower downstream performance than prompting in English. Figure 3 shows the performance difference between English and in-language prompting across models, languages and tasks for the best two performing cross-lingual vocabulary adaptation methods (Heuristics and CLP+).

Surprisingly, we observe no major performance drop with in-language prompting in the majority of the zero-shot settings across languages. We note similar or better performance with in-language prompting in 10 out of 16 settings.We also observe a drop of 0.03 or larger (non-shaded areas in Figure 3) in only 5 out of 16 settings. The few-shot settings also exhibited similar trends with substantial performance degradation of 0.03 or more in 5 out of 12 settings, and similar or better performance in 8 out of 12 settings. Some in-language prompting cases with lower performance than English, such as in German across tasks and zero-shot NLI in Arabic and Swahili, can be related to the tokenization effects discussed in §5.1. Previous studies (Rust et al., 2021; Bostrom and Durrett, 2020; Fujii et al., 2023) have also found a strong correlation between tokenization and performance. Further investigation is needed on how to mitigate performance degradation, especially in few-shot settings such as Japanese NLI and Arabic SPAN.

LAPT Steps. LAPT is an integral part of recent cross-lingual vocabulary adaptation methods (Minixhofer et al., 2022; Dobler and de Melo, 2023; Ostendorff and Rehm, 2023; Downey et al., 2023). However, it is a computationally intensive task that requires loading and updating models with billions of parameters over a large number of training steps. Therefore, we investigate the relationship between downstream performance and the number of LAPT steps, i.e. every 2k steps starting from

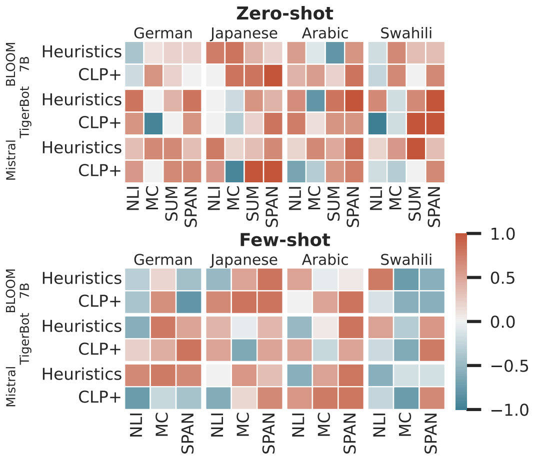

Figure 4: Kendall's τ correlation between the number of LAPT steps and performance (in-language prompting).

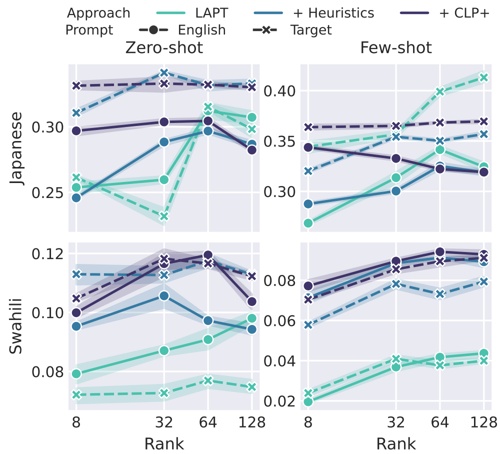

Figure 5: SPAN performance changes with respect to LoRA rank r.

1k and every 10k after 13k. Figure 4 shows the Kendall's tau (Kendall, 1938) correlation coefficients between LAPT steps and performance.

Overall, LAPT helps improve downstream performance in both zero- and few-shot settings in 69.5% and 59.4% of the cases, respectively.[7] In particular, both TigerBot and Mistral, which are not as multilingual as BLOOM, tend to benefit more from LAPT, especially in zero-shot SUM and SPAN across languages. This suggests that LAPT helps source LLMs to supplement target language knowledge to reach competitive performance with BLOOM 7B in a similar number of steps.

Next, we examine the correlation coefficients by task across zero- and few-shot settings. We often observe negative or no correlation in zero-shot MC across languages, zero-shot NLI in Japanese and Swahili, few-shot NLI in German, Japanese and Swahili, few-shot MC in Swahili, and few-shot SPAN in German and Swahili. In contrast, SPAN and SUM generally benefit well from LAPT in zero-shot settings across languages, in addition to few-shot SPAN in Japanese and Arabic. We hypothesize that because zero-shot tasks, especially SUM and SPAN, can be more challenging than the other text classification tasks (Davletov et al., 2021; Yamaguchi et al., 2022), they may require better target language representations to perform well.

LoRA Rank r. There is a trade-off between computational efficiency and performance when adapting LLMs with LoRA (Hu et al., 2021). We analyze how the LoRA rank affects performance in cross-lingual vocabulary adaptation. To keep computational costs low, we experiment by setting $r = \{8, 32, 64, 128\}$ using BLOOM-1B on SPAN in Japanese and Swahili where we observed large performance variations (Table 2). Figure 5 shows how performance changes with respect to r.

In general, the performance of target models does not increase with r in the zero-shot setting. The only exception is Heuristics in Japanese with English prompting, i.e. from 0.246 ($r = 8$) to 0.297 ($r = 64$). We observe that performance improves with r in the few-shot setting. CLP+ in Japanese with English prompting is an exception, i.e. from 0.344 ($r = 8$) to 0.319 ($r = 128$). This suggests that setting $r = 8$ is a good option in zero-shot settings. Increasing r to 32, 64 or 128 can yield better few-shot performance but results to higher computational costs. For instance, the best-performing Swahili model ($r = 64$) results in a 14% increase in the number of trainable parameters compared to $r = 8$. Careful cost-benefit consideration is needed to choose an optimal r.

7 Conclusion

We investigated the effectiveness of cross-lingual vocabulary adaptation on LLM inference efficiency. Our extensive experiments in four diverse languages demonstrated that cross-lingual vocabulary adaptation substantially contributes to LLM inference speedups of up to 271.5% while maintaining comparable downstream performance to baselines when adapting multilingual LLMs. In future work, we will explore various inference-aware methods for cross-lingual transfer, such as cost-effective subword vocabulary selection (Gee et al., 2023).

[7] We observe similar trends with English prompting (Figure 6 in the Appendix).

Limitations

Prompt Tuning. We use a translated version of in-language prompts from English. This may affect the downstream performance due to machine translation noise, underestimating the performance of in-language prompting.

Languages. Although this study covers four linguistically diverse languages (German, Arabic, Japanese, and Swahili), it is an interesting study for future work to assess more languages.

Model Size. This paper considers LLMs of various sizes ranging from 1B to 7B, which are far larger than those tested in previous cross-lingual vocabulary adaptation studies. Note that inference efficiency measured by the number of processed/generated tokens is not affected by the model size. However, investigating the performance of cross-lingual vocabulary adaptation approaches with larger models would be valuable in future studies.

References

Orevaoghene Ahia, Sachin Kumar, Hila Gonen, Jungo Kasai, David Mortensen, Noah Smith, and Yulia Tsvetkov. 2023. Do all languages cost the same? tokenization in the era of commercial language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9904–9923, Singapore. Association for Computational Linguistics.

Kabir Ahuja, Harshita Diddee, Rishav Hada, Millicent Ochieng, Krithika Ramesh, Prachi Jain, Akshay Nambi, Tanuja Ganu, Sameer Segal, Mohamed Ahmed, Kalika Bali, and Sunayana Sitaram. 2023. MEGA: Multilingual evaluation of generative AI. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4232–4267, Singapore. Association for Computational Linguistics.

Rami Al-Rfou, Dokook Choe, Noah Constant, Mandy Guo, and Llion Jones. 2019. Character-level language modeling with deeper self-attention. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):3159–3166.

Mehdi Ali, Michael Fromm, Klaudia Thellmann, Richard Rutmann, Max Lübbering, Johannes Leveling, Katrin Klug, Jan Ebert, Niclas Doll, Jasper Schulze Buschhoff, Charvi Jain, Alexander Arno Weber, Lena Jurkschat, Hammam Abdelwahab, Chelsea John, Pedro Ortiz Suarez, Malte Ostendorff, Samuel Weinbach, Rafet Sifa, Stefan Kesselheim, and Nicolas Flores-Herr. 2023. Tokenizer choice for LLM training: Negligible or crucial? *ArXiv*, abs/2310.08754.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2021. AraGPT2: Pre-trained transformer for Arabic language generation. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 196–207, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Abhinand Balachandran. 2023. Tamil-LLaMA: A new Tamil language model based on LLaMA 2. *ArXiv*, abs/2311.05845.

Yoshua Bengio, Réjean Ducharme, and Pascal Vincent. 2000. A neural probabilistic language model. In *Advances in Neural Information Processing Systems*, volume 13. MIT Press.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Kaj Bostrom and Greg Durrett. 2020. Byte pair encoding is suboptimal for language model pretraining. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4617–4624, Online. Association for Computational Linguistics.

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 2020*, pages 1324–1334, Online. Association for Computational Linguistics.

Ye Chen, Wei Cai, Liangmin Wu, Xiaowei Li, Zhanxuan Xin, and Cong Fu. 2023. TigerBot: An open multilingual multitask LLM. *ArXiv*, abs/2312.08688.

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 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics.

Yiming Cui, Ziqing Yang, and Xin Yao. 2023. Efficient and effective text encoding for Chinese LLaMA and Alpaca. *ArXiv*, abs/2304.08177.

Adis Davletov, Nikolay Arefyev, Denis Gordeev, and Alexey Rey. 2021. LIORI at SemEval-2021 task 2: Span prediction and binary classification approaches to word-in-context disambiguation. In *Proceedings* of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 780–786, Online. Association for Computational Linguistics.

Wietse de Vries and Malvina Nissim. 2021. As good as new. how to successfully recycle English GPT-2 to make models for other languages. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 836–846, Online. Association for Computational Linguistics.

Konstantin Dobler and Gerard de Melo. 2023. FOCUS: Effective embedding initialization for monolingual specialization of multilingual models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13440–13454, Singapore. Association for Computational Linguistics.

C.m. Downey, Terra Blevins, Nora Goldfine, and Shane Steinert-Threlkeld. 2023. Embedding structure matters: Comparing methods to adapt multilingual vocabularies to new languages. In *Proceedings of the 3rd Workshop on Multi-lingual Representation Learning (MRL)*, pages 268–281, Singapore. Association for Computational Linguistics.

Takuro Fujii, Koki Shibata, Atsuki Yamaguchi, Terufumi Morishita, and Yasuhiro Sogawa. 2023. How do different tokenizers perform on downstream tasks in scriptio continua languages?: A case study in Japanese. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, pages 39–49, Toronto, Canada. Association for Computational Linguistics.

Leonidas Gee, Leonardo Rigutini, Marco Ernandes, and Andrea Zugarini. 2023. Multi-word tokenization for sequence compression. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 612–621, Singapore. Association for Computational Linguistics.

Edward Gow-Smith, Harish Tayyar Madabushi, Carolina Scarton, and Aline Villavicencio. 2022. Improving tokenisation by alternative treatment of spaces. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11430–11443, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Tahmid Hasan, Abhik Bhattacharjee, Md. Saiful Islam, Kazi Mubasshir, 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-IJCNLP 2021*, pages 4693–4703, Online. Association for Computational Linguistics.

Valentin Hofmann, Hinrich Schuetze, and Janet Pierrehumbert. 2022. An embarrassingly simple method to mitigate undesirable properties of pretrained language model tokenizers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 385–393, Dublin, Ireland. Association for Computational Linguistics.

J. Edward Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. 2021. LoRA: Low-rank adaptation of large language models. *ArXiv*, abs/2106.09685.

Timm Jansen, Yangling Tong, Victoria Zevallos, and Pedro Ortiz Suarez. 2022. Perplexed by quality: A perplexity-based method for adult and harmful content detection in multilingual heterogeneous web data. *ArXiv*, abs/2212.10440.

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

M. G. Kendall. 1938. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93.

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.

Kentaro Kurihara, Daisuke Kawahara, and Tomohide Shibata. 2022. JGLUE: Japanese general language understanding evaluation. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 2957–2966, Marseille, France. European Language Resources Association.

Celio H. N. Larcher, Marcos Piau, Paulo Finardi, Pedro Gengo, Piero Esposito, and Vinicius F. Carid'a. 2023. Cabrita: Closing the gap for foreign languages. *ArXiv*, abs/2308.11878.

Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Šaško, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angelina McMillan-Major, Philipp Schmid, Sylvain Gugger, Clément Delangue, Théo Matussière, Lysandre Debut, Stas Bekman, Pierric Cistac, Thibault Goehringer, Victor Mustar, François Lagunas, Alexander Rush, and Thomas Wolf. 2021. Datasets: A community library for natural language processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 175–184, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Bill Yuchen Lin, Seyeon Lee, Xiaoyang Qiao, and Xiang Ren. 2021. Common sense beyond English: Evaluating and improving multilingual language models for commonsense reasoning. 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 1274–1287, Online. Association for Computational Linguistics.

Xi Victoria Lin, Todor Mihaylov, Mikel Artetxe, Tianlu Wang, Shuohui Chen, Daniel Simig, Myle Ott, Naman Goyal, Shruti Bhosale, Jingfei Du, Ramakanth Pasunuru, Sam Shleifer, Punit Singh Koura, Vishrav Chaudhary, Brian O'Horo, Jeff Wang, Luke Zettlemoyer, Zornitsa Kozareva, Mona Diab, Veselin Stoyanov, and Xian Li. 2022. Few-shot learning with multilingual generative language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9019–9052, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, and Benjamin Bossan. 2022. PEFT: State-of-the-art parameter-efficient fine-tuning methods. `https://github.com/huggingface/peft`.

Andre Martins and Ramon Astudillo. 2016. From softmax to sparsemax: A sparse model of attention and multi-label classification. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1614–1623, New York, New York, USA. PMLR.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

Benjamin Minixhofer, Fabian Paischer, and Navid Rekabsaz. 2022. WECHSEL: Effective initialization of subword embeddings for cross-lingual transfer of monolingual language models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3992–4006, Seattle, United States. Association for Computational Linguistics.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng Xin Yong, Hailey Schoelkopf, Xiangru Tang, Dragomir Radev, Alham Fikri Aji, Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. 2023. Crosslingual generalization through multitask finetuning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15991–16111, Toronto, Canada. Association for Computational Linguistics.

Benjamin Muller, Antonios Anastasopoulos, Benoît Sagot, and Djamé Seddah. 2021. When being unseen from mBERT is just the beginning: Handling new languages with multilingual language models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 448–462, Online. Association for Computational Linguistics.

OpenAI. 2023. GPT-4 technical report. *ArXiv*, abs/2303.08774.

Malte Ostendorff and Georg Rehm. 2023. Efficient language model training through cross-lingual and progressive transfer learning. *ArXiv*, abs/2301.09626.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Aleksandar Petrov, Emanuele La Malfa, Philip Torr, and Adel Bibi. 2023. Language model tokenizers introduce unfairness between languages. In *Thirty-seventh Conference on Neural Information Processing Systems*.

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.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ili'c, Daniel Hesslow, Roman Castagn'e, Alexandra Sasha Luccioni, Franccois Yvon, Matthias Gallé, Jonathan Tow, Alexander M. Rush, Stella Biderman, Albert Webson, Pawan Sasanka Ammanamanchi, Thomas Wang, Benoît Sagot, Niklas Muennighoff, Albert Villanova del Moral, Olatunji Ruwase, Rachel Bawden, Stas Bekman, Angelina McMillan-Major, Iz Beltagy, Huu Nguyen, Lucile Saulnier, Samson Tan, Pedro Ortiz Suarez, Victor Sanh, Hugo Laurenccon, Yacine Jernite, Julien Launay, Margaret Mitchell, Colin Raffel, Aaron Gokaslan, Adi Simhi, Aitor Soroa Etxabe, Alham Fikri Aji, Amit Alfassy, Anna Rogers, Ariel Kreisberg Nitzav, Canwen Xu, Chenghao Mou, Chris C. Emezue, Christopher Klamm, Colin Leong, Daniel Alexander van Strien, David Ifeoluwa Adelani, Dragomir R. Radev, Eduardo Gonz'alez Ponferrada, Efrat Levkovizh, Ethan Kim, Eyal

Natan, Francesco De Toni, Gérard Dupont, Germán Kruszewski, Giada Pistilli, Hady ElSahar, Hamza Benyamina, Hieu Trung Tran, Ian Yu, Idris Abdulmumin, Isaac Johnson, Itziar Gonzalez-Dios, Javier de la Rosa, Jenny Chim, Jesse Dodge, Jian Zhu, Jonathan Chang, Jorg Frohberg, Josephine L. Tobing, Joydeep Bhattacharjee, Khalid Almubarak, Kimbo Chen, Kyle Lo, Leandro von Werra, Leon Weber, Long Phan, Loubna Ben Allal, Ludovic Tanguy, Manan Dey, Manuel Romero Muñoz, Maraim Masoud, María Grandury, Mario vSavsko, Max Huang, Maximin Coavoux, Mayank Singh, Mike Tian-Jian Jiang, Minh Chien Vu, Mohammad Ali Jauhar, Mustafa Ghaleb, Nishant Subramani, Nora Kassner, Nurulaqilla Khamis, Olivier Nguyen, Omar Espejel, Ona de Gibert, Paulo Villegas, Peter Henderson, Pierre Colombo, Priscilla Amuok, Quentin Lhoest, Rheza Harliman, Rishi Bommasani, Roberto L'opez, Rui Ribeiro, Salomey Osei, Sampo Pyysalo, Sebastian Nagel, Shamik Bose, Shamsuddeen Hassan Muhammad, Shanya Sharma Sharma, S. Longpre, So maieh Nikpoor, S. Silberberg, Suhas Pai, Sydney Zink, Tiago Timponi Torrent, Timo Schick, Tristan Thrush, Valentin Danchev, Vassilina Nikoulina, Veronika Laippala, Violette Lepercq, Vrinda Prabhu, Zaid Alyafeai, Zeerak Talat, Arun Raja, Benjamin Heinzerling, Chenglei Si, Elizabeth Salesky, Sabrina J. Mielke, Wilson Y. Lee, Abheesht Sharma, Andrea Santilli, Antoine Chaffin, Arnaud Stiegler, Debajyoti Datta, Eliza Szczechla, Gunjan Chhablani, Han Wang, Harshit Pandey, Hendrik Strobelt, Jason Alan Fries, Jos Rozen, Leo Gao, Lintang Sutawika, M Saiful Bari, Maged S. Al-Shaibani, Matteo Manica, Nihal V. Nayak, Ryan Teehan, Samuel Albanie, Sheng Shen, Srulik Ben-David, Stephen H. Bach, Taewoon Kim, Tali Bers, Thibault Févry, Trishala Neeraj, Urmish Thakker, Vikas Raunak, Xiang Tang, Zheng-Xin Yong, Zhiqing Sun, Shaked Brody, Y Uri, Hadar Tojarieh, Adam Roberts, Hyung Won Chung, Jaesung Tae, Jason Phang, Ofir Press, Conglong Li, Deepak Narayanan, Hatim Bourfoune, Jared Casper, Jeff Rasley, Max Ryabinin, Mayank Mishra, Minjia Zhang, Mohammad Shoeybi, Myriam Peyrounette, Nicolas Patry, Nouamane Tazi, Omar Sanseviero, Patrick von Platen, Pierre Cornette, Pierre Franccois Lavall'ee, Rémi Lacroix, Samyam Rajbhandari, Sanchit Gandhi, Shaden Smith, Stéphane Requena, Suraj Patil, Tim Dettmers, Ahmed Baruwa, Amanpreet Singh, Anastasia Cheveleva, Anne-Laure Ligozat, Arjun Subramonian, Aur'elie N'ev'eol, Charles Lovering, Daniel H Garrette, Deepak R. Tunuguntla, Ehud Reiter, Ekaterina Taktasheva, Ekaterina Voloshina, Eli Bogdanov, Genta Indra Winata, Hailey Schoelkopf, Jan-Christoph Kalo, Jekaterina Novikova, Jessica Zosa Forde, Xiangru Tang, Jungo Kasai, Ken Kawamura, Liam Hazan, Marine Carpuat, Miruna Clinciu, Najoung Kim, Newton Cheng, Oleg Serikov, Omer Antverg, Oskar van der Wal, Rui Zhang, Ruochen Zhang, Sebastian Gehrmann, Shachar Mirkin, S. Osher Pais, Tatiana Shavrina, Thomas Scialom, Tian Yun, Tomasz Limisiewicz, Verena Rieser, Vitaly Protasov, Vladislav Mikhailov, Yada Pruksachatkun, Yonatan Belinkov, Zachary Bamberger, Zdenvek Kasner, Zdeněk Kasner, Amanda Pestana, Amir Feizpour, Ammar Khan, Amy Faranak, Ananda Santa Rosa Santos, Anthony Hevia, Antigona Unldreaj, Arash Aghagol, Arezoo Abdollahi, Aycha Tammour, Azadeh HajiHosseini, Bahareh Behroozi, Benjamin Ayoade Ajibade, Bharat Kumar Saxena, Carlos Muñoz Ferrandis, Danish Contractor, David M. Lansky, Davis David, Douwe Kiela, Duong Anh Nguyen, Edward Tan, Emi Baylor, Ezinwanne Ozoani, Fatim Tahirah Mirza, Frankline Ononiwu, Habib Rezanejad, H.A. Jones, Indrani Bhattacharya, Irene Solaiman, Irina Sedenko, Isar Nejadgholi, Jan Passmore, Joshua Seltzer, Julio Bonis Sanz, Karen Fort, Lívia Dutra, Mairon Samagaio, Maraim Elbadri, Margot Mieskes, Marissa Gerchick, Martha Akinlolu, Michael McKenna, Mike Qiu, Muhammed Ghauri, Mykola Burynok, Nafis Abrar, Nazneen Rajani, Nour Elkott, Nourhan Fahmy, Olanrewaju Samuel, Ran An, R. P. Kromann, Ryan Hao, Samira Alizadeh, Sarmad Shubber, Silas L. Wang, Sourav Roy, Sylvain Viguier, Thanh-Cong Le, Tobi Oyebade, Trieu Nguyen Hai Le, Yoyo Yang, Zachary Kyle Nguyen, Abhinav Ramesh Kashyap, Alfredo Palasciano, Alison Callahan, Anima Shukla, Antonio Miranda-Escalada, Ayush Kumar Singh, Benjamin Beilharz, Bo Wang, Caio Matheus Fonseca de Brito, Chenxi Zhou, Chirag Jain, Chuxin Xu, Clémentine Fourrier, Daniel Le'on Perin'an, Daniel Molano, Dian Yu, Enrique Manjavacas, Fabio Barth, Florian Fuhrimann, Gabriel Altay, Giyaseddin Bayrak, Gully Burns, Helena U. Vrabec, Iman I.B. Bello, Isha Dash, Ji Soo Kang, John Giorgi, Jonas Golde, Jose David Posada, Karthi Sivaraman, Lokesh Bulchandani, Lu Liu, Luisa Shinzato, Madeleine Hahn de Bykhovetz, Maiko Takeuchi, Marc Pàmies, María Andrea Castillo, Marianna Nezhurina, Mario Sanger, Matthias Samwald, Michael Cullan, Michael Weinberg, M Wolf, Mina Mihaljcic, Minna Liu, Moritz Freidank, Myungsun Kang, Natasha Seelam, Nathan Dahlberg, Nicholas Michio Broad, Nikolaus Muellner, Pascale Fung, Patricia Haller, R. Chandrasekhar, Renata Eisenberg, Robert Martin, Rodrigo L. Canalli, Rosaline Su, Ruisi Su, Samuel Cahyawijaya, Samuele Garda, Shlok S Deshmukh, Shubhanshu Mishra, Sid Kiblawi, Simon Ott, Sinee Sang-aroonsiri, Srishti Kumar, Stefan Schweter, Sushil Pratap Bharati, Tanmay Laud, Th'eo Gigant, Tomoya Kainuma, Wojciech Kusa, Yanis Labrak, Yashasvi Bajaj, Y. Venkatraman, Yifan Xu, Ying Xu, Yu Xu, Zhee Xao Tan, Zhongli Xie, Zifan Ye, Mathilde Bras, Younes Belkada, and Thomas Wolf. 2022. BLOOM: A 176b-parameter open-access multilingual language model. *ArXiv*, abs/2211.05100.

Mike Schuster and Kaisuke Nakajima. 2012. Japanese and Korean voice search. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5149–5152.

Thomas Scialom, Paul-Alexis Dray, Sylvain Lamprier, Benjamin Piwowarski, and Jacopo Staiano. 2020. MLSUM: The multilingual summarization corpus. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8051–8067, Online. 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.

Jimin Sun, Patrick Fernandes, Xinyi Wang, and Graham Neubig. 2023. A multi-dimensional evaluation of tokenizer-free multilingual pretrained models. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1725–1735, Dubrovnik, Croatia. Association for Computational Linguistics.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Cagri Toraman, Eyup Halit Yilmaz, Furkan Şahinuç, and Oguzhan Ozcelik. 2023. Impact of tokenization on language models: An analysis for Turkish. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 22(4).

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. LLaMA: Open and efficient foundation language models. *ArXiv*, abs/2302.13971.

Hugo Touvron, Louis Martin, Kevin R. Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Daniel M. Bikel, Lukas Blecher, Cristian Cantón Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony S. Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel M. Kloumann, A. V. 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, R. Subramanian, Xia Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zhengxu Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. LLaMA 2: Open foundation and fine-tuned chat models. *ArXiv*, abs/2307.09288.

Ke M. Tran. 2019. From English to foreign languages: Transferring pre-trained language models. *ArXiv*, abs/2002.07306.

Asahi Ushio, Yi Zhou, and Jose Camacho-Collados. 2023. Efficient multilingual language model compression through vocabulary trimming. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 14725–14739, Singapore. Association for Computational Linguistics.

Zihan Wang, Karthikeyan K, Stephen Mayhew, and Dan Roth. 2020. Extending multilingual BERT to low-resource languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2649–2656, Online. Association for Computational Linguistics.

Barack W. Wanjawa, Lilian D. A. Wanzare, Florence Indede, Owen Mconyango, Lawrence Muchemi, and Edward Ombui. 2023. KenSwQuAD—A question answering dataset for swahili low-resource language. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 22(4).

Miles Williams and Nikolaos Aletras. 2023. Frustratingly simple memory efficiency for pre-trained language models via dynamic embedding pruning. *ArXiv*, abs/2309.08708.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

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

Atsuki Yamaguchi, Gaku Morio, Hiroaki Ozaki, and Yasuhiro Sogawa. 2022. Hitachi at SemEval-2022 task 2: On the effectiveness of span-based classification approaches for multilingual idiomaticity detection. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 135–144, Seattle, United States. Association for Computational Linguistics.

Zheng Xin Yong, Hailey Schoelkopf, Niklas Muennighoff, Alham Fikri Aji, David Ifeoluwa Adelani, Khalid Almubarak, M Saiful Bari, Lintang Sutawika, Jungo Kasai, Ahmed Baruwa, Genta Winata, Stella Biderman, Edward Raff, Dragomir Radev, and Vassilina Nikoulina. 2023. BLOOM+1: Adding language support to BLOOM for zero-shot prompting.

In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11682–11703, Toronto, Canada. Association for Computational Linguistics.

Appendix

A Implementation Details

A.1 Tokenizer

To reduce the computational costs, we utilized publicly available existing tokenizers for each target language, which means we used them as $\mathcal{T}_t$. Table 3 lists the tokenizers used in our experiments.

A.2 Language-specific pre-trained LM

For language-specific pre-trained LMs used in CLP and CLP+, we used the corresponding models to $\mathcal{T}_t$, which are listed in Table 3 and are all decoder-based models. Note that we only used the embedding of each language-specific pre-trained LM for vocabulary adaptation, and therefore, one can also use encoder and encoder-decoder based models.

A.3 fastText in FOCUS

For FOCUS, we trained a fastText model for each language on a corresponding CC-100 (Conneau et al., 2020) text with the same configuration as Dobler and de Melo (2023).

A.4 Hyperparameters and Generation Configurations

LAPT Table 4 shows the hyperparameters in LAPT for each model size. Note that due to the computational resource constraints and funds for running experiments, we could run pre-training of up to four days for each approach. Therefore, we picked up checkpoints with the largest number of steps available across models with the same base model (i.e. BLOOM-1B, BLOOM-7B, etc.) and language for evaluation to make a fair comparison. We also temporarily trimmed the unused embeddings of BLOOM models for LAPT, whose tokens did not appear in the training corpus during pre-training to save memory and for faster computation.[8]

Generation Following Cui et al. (2023), we introduced a verbalizer for the classification tasks: NLI and MC, where we mapped the first generated token into a label to compute accuracy. For mapping, we simply picked up a token with the maximum log-likelihood among candidate tokenized words. The list of candidate label words for each task is shown in Table 5. Table 6 lists the parameters used during evaluation. To make a fair comparison, we did not conduct any generation parameter tuning and used the same ones across all approaches. For SUM and few-shot SPAN in Swahili, we truncated an article whenever it exceeded the maximum prompt length of 4,096 to avoid the CUDA out-of-memory error.

[8] We used the implementations by Ushio et al. (2023) and Williams and Aletras (2023).

A.5 Checkpoints

As explained in A.4, we trained all models for up to four days each due to limited computational resources and funds for experiments. The only exception was Swahili since the dataset is small enough to complete LAPT. To make a fair comparison, we used checkpoints with the largest number of steps available across models with the same target language and base model. Table 7 shows the list of checkpoints used for evaluation.

Model	Language			
	de	ja	ar	sw
BLOOM-1B	47k	48k	50k	9k
BLOOM-7B	8k	8k	8k	4k
TigerBot-7B	8k	8k	8k	4k
Mistral-7B	6k	6k	6k	4k

Table 7: List of checkpoints used for evaluation. We used checkpoints with the largest number of steps available across all models with the same base model and language.

A.6 Libraries

We implement our models using PyTorch (Paszke et al., 2019), Hugging Face Transformers (Wolf et al., 2020) and PEFT (Mangrulkar et al., 2022). We preprocess data with Hugging Face Datasets (Lhoest et al., 2021). For evaluation, we use Hugging Face Evaluate[9] to compute downstream performance metrics.

A.7 Prompt Templates

Table 8 shows the prompt templates used in our evaluation.

A.7.1 Code

The anonymized code is available as supplementary material in the submission for reference.

B Licenses

This study used various publicly available models and datasets with different licenses, as detailed below, all of which permit their use for academic research.

[9] https://github.com/huggingface/evaluate

Language	Tokenization Algorithm	Hugging Face Identifier	Citation	License
German	Byte-level BPE	malteos/gpt2-xl-wechsel-german		MIT
Japanese	Unigram	rinna/japanese-gpt-neox-3.6b-instruction-ppo		MIT
Arabic	Byte-level BPE	aubmindlab/aragpt2-base	(Antoun et al., 2021)	See here
Swahili	Byte-level BPE	benjamin/gpt2-wechsel-swahili	(Minixhofer et al., 2022)	MIT

Table 3: List of tokenizers used for each language-specific model with vocabulary adaptation.

Hyperparameters	1B	7B
Batch size	8	16
Gradient accumulation steps	4	4
Maximum number of training epochs	1	1
Maximum number of training days	4	4
Adam ϵ	1e-8	1e-8
Adam β_1	0.9	0.9
Adam β_2	0.999	0.999
Sequence length	1,024	1,024
Learning rate	1e-4	1e-4
Learning rate scheduler	cosine	cosine
Warmup steps	100	100
Weight decay	0.01	0.01
Attention dropout	0.0	0.0
Dropout	0.05	0.05
LoRA rank r	8	8
LoRA dropout	0.05	0.05
LoRA α	32	32
Training precision	FP16	FP16
Model quantization	int 8	int 8

Table 4: Hyperparameters for LAPT.

Task	Language	Label words
NLI	English	True, False, Neither
	German	Wahr, Falsch, Weder
	Japanese	真，偽，どちらでもない
	Arabic	صحيح, خطأ, لا هذا ولا ذاك
	Swahili	Kweli, Uongo, Wala
MC	All	A, B, C, D, E

Table 5: List of candidate label words for each classification task.

B.1 Models

BLOOM is licensed under the BigScience RAIL License.[10] TigerBot and Mistral are licensed under the Apache-2.0 License. The licenses of the helper models are listed in Table 3.

B.2 Datasets

XNLI is distributed under CC BY-NC 4.0. JNLI, XQuAD, and JSQuAD are distributed under CC BY-SA 4.0. XCSQA is a derivative of CommonsenseQA (Talmor et al., 2019), which is licensed under an MIT license. OSCAR and KenSwQuAD are licensed under CC0 – no rights reserved. XL-Sum is licensed under CC BY-NC-SA 4.0, while MLSUM is distributed under an MIT license.

[10] https://huggingface.co/spaces/bigscience/license

Parameters	Values
Maximum prompt length	4,096
Temperature	0.8
Repetition penalty	1.1
Top k	40
Top p	0.9
Beam width	5
Sampling	True
Early stopping	True

Table 6: Parameters for generation.

C Results

C.1 Additional Results

Table 9 shows the results with standard deviations when prompted in English, and Table 10 shows the results with standard deviations when prompted in a target language.

C.2 English Downstream Performance

Table 11 shows the results on the English datasets. Despite the entire replacement of embeddings for cross-lingual vocabulary adaptation approaches, their adapted models exhibit comparable or better results in most of the tasks for BLOOM, except for SPAN, where Source showed the best result followed by LAPT. This can be ascribed to the following reasons: First, LAPT can retain more source model knowledge than cross-lingual vocabulary adaptation approaches, as their embeddings have not changed. Second, SPAN can be seen as a challenging task as it requires more linguistic understanding of a prompt than simply classifying a text as in NLI and MC. We, therefore, see such a huge performance difference in SPAN since cross-lingual vocabulary adaptation approaches lost more

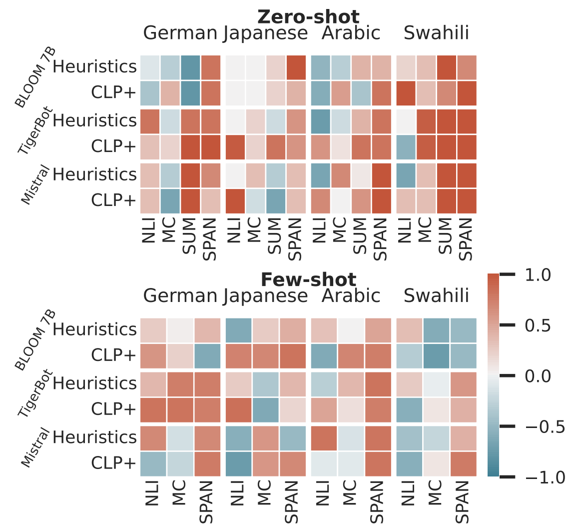

Figure 6: Kendall's τ correlation between the number of LAPT steps and performance (English prompting).

source linguistic knowledge than LAPT counterparts in exchange for faster inference in a target language.

For TigerBot and Mistral, which are not as multilingual as BLOOM, we see quite similar trends observed in §5.2 in that (1) models with cross-lingual vocabulary adaptation fail to achieve competitive downstream performance to the baselines and (2) their few-shot performances are far lower than LAPT. These results suggest that there can be a relationship between downstream performances in a target language and those in English, and maintaining competitive downstream performance to the baselines in English might be a key to improving models with cross-lingual vocabulary adaptation in terms of their downstream performance.

C.3 How helpful is LAPT for LLMs with cross-lingual vocabulary adaptation?

Figure 6 visualizes Kendall's tau correlation coefficient between the number of LAPT steps and downstream performance when prompted in English. Similar to Figure 4, we observe that LAPT helped improve downstream performance in both zero-shot and few-shot settings even when prompted in a target language in 65.6% and 63.5% of the cases, respectively.

C.4 Loss Curves

Figures 7 to 10 show the loss curves in LAPT for each model setting.

Approach	Japanese		Swahili	
	English	Target	English	Target
		Zero-shot		
LAPT	0.720	0.333	0.788	0.187
+ Heuristics	0.453	0.173	0.173	0.160
+ CLP+	-0.160	-0.106	-0.226	0.066
		Few-shot		
LAPT	0.626	1.00	0.453	0.906
+ Heuristics	0.706	0.600	0.591	0.701
+ CLP+	-0.946	0.626	0.886	0.756

Table 12: Kendall's tau correlation coefficients corresponding to Figure 5. We include LAPT results for reference.

C.5 Kendall's Tau Correlation Coefficient for Figure 5

Table 12 lists all Kendall's tau correlation coefficients corresponding to Figure 5 in §6. Models using cross-lingual vocabulary adaptation do not exhibit a strong correlation in the zero-shot setting, ranging from -0.226 to 0.173. The only exception is Heuristics in Japanese with English prompting (0.45). We observe a positive correlation ranging (0.59-0.889) in the few-shot setting, except for CLP+ in Japanese with English prompting (-0.95).

Task	Language	Template
NLI	English	{premise} Question: {hypothesis} True, False, or Neither? Answer:
	German	{premise} Frage: {hypothesis} Wahr, Falsch oder Weder? Antwort:
	Japanese	{premise} 質問: {hypothesis} 真、偽、どちらでもない？ 答え:
	Arabic	{premise} سؤال: {hypothesis} صحيح ، خطأ أو لا هذا ولا ذاك؟ إجابة:
	Swahili	{premise} Swali: {hypothesis} Kweli, Uongo au Wala? Jibu:
MC	English	{question} A. {choice_1}, B. {choice_2}, C. {choice_3}, D. {choice_4}, E. {choice_5} Answer:
	German	{question} A. {choice_1}, B. {choice_2}, C. {choice_3}, D. {choice_4}, E. {choice_5} Antwort:
	Japanese	{question} A. {choice_1}, B. {choice_2}, C. {choice_3}, D. {choice_4}, E. {choice_5} 答え:
	Arabic	{question} A. {choice_1}, B. {choice_2}, C. {choice_3}, D. {choice_4}, E. {choice_5} إجابة:
	Swahili	{question} A. {choice_1}, B. {choice_2}, C. {choice_3}, D. {choice_4}, E. {choice_5} Jibu:
SUM	English	Write a short summary of the following text in {language}. Article: {text} Summary:
	German	Schreiben Sie eine kurze Zusammenfassung des folgenden Textes auf Deutsch. Artikel: {text} Zusammenfassung:
	Japanese	次の文章の要約を日本語で書きなさい。記事: {text} 要約:
	Arabic	اكتب ملخصًا قصيرًا للنص التالي باللغة العربية. المقالة: {text} الملخص:
	Swahili	Andika muhtasari mfupi wa maandishi yafuatayo kwa Kiswahili. Makala: {text} Muhtasari:
SPAN	English	Answer the following question. Context: {context} Question: {question} Answer:
	German	Beantworten Sie die folgende Frage. Artikel: {context} Frage: {question} Antwort:
	Japanese	次の文章の質問に答えなさい。文章: {context} 質問: {question} 答え:
	Arabic	أجب على السؤال التالي. سياق: {context} السؤال: {question} الإجابة:
	Swahili	Jibu swali lifuatalo. Makala: {context} Swali: {question} Jibu:

Table 8: Prompt template for each task and language.

Approach	German				Japanese				Arabic				Swahili			
	NLI	MC	SUM	SPAN	NLI	MC	SUM	SPAN	NLI	MC	SUM	SPAN	NLI	MC	SUM	SPAN
BLOOM-1B							**Zero-shot**									
Source	.34$_{.00}$	.20$_{.00}$	14.9$_{0.7}$	.08$_{.00}$	.17$_{.00}$	.25$_{.00}$	3.7$_{0.1}$	.23$_{.00}$	.35$_{.00}$	.19$_{.00}$	9.9$_{0.1}$	.17$_{.01}$	.34$_{.00}$	.21$_{.00}$	10.6$_{0.4}$	.08$_{.00}$
LAPT	.33$_{.00}$	.21$_{.01}$	17.0$_{0.1}$	.09$_{.01}$	.17$_{.00}$	.20$_{.00}$	17.2$_{0.3}$	.25$_{.00}$	.34$_{.00}$	.18$_{.00}$	11.6$_{0.1}$	.14$_{.01}$	.34$_{.00}$	.18$_{.00}$	9.7$_{0.1}$	.08$_{.00}$
+ Random	.31$_{.00}$	**.22**$_{.00}$	**19.0**$_{0.2}$	**.12**$_{.00}$	**.17**$_{.00}$	.20$_{.00}$	**17.7**$_{0.2}$	.19$_{.00}$	**.36**$_{.00}$	.18$_{.00}$	10.4$_{0.2}$	.09$_{.00}$	.32$_{.00}$	**.22**$_{.00}$	9.8$_{0.1}$	.05$_{.00}$
+ Heuristics	.32$_{.00}$	**.22**$_{.00}$	**17.7**$_{0.4}$	**.11**$_{.00}$	**.17**$_{.00}$	.20$_{.00}$	**17.9**$_{0.1}$	**.25**$_{.00}$	**.39**$_{.00}$	.18$_{.00}$	10.9$_{0.1}$	.13$_{.00}$	**.35**$_{.00}$	**.22**$_{.00}$	**11.9**$_{0.1}$	**.10**$_{.00}$
+ CLP	**.34**$_{.00}$	**.23**$_{.00}$	**18.7**$_{0.2}$	**.12**$_{.01}$	**.17**$_{.00}$	.21$_{.00}$	**17.6**$_{0.1}$	**.27**$_{.01}$	**.36**$_{.00}$	.18$_{.00}$	11.2$_{0.1}$	.14$_{.00}$	**.38**$_{.00}$	**.22**$_{.00}$	**11.6**$_{0.1}$	**.11**$_{.00}$
+ FOCUS	**.34**$_{.00}$	**.22**$_{.00}$	**17.4**$_{0.4}$	**.11**$_{.00}$	**.17**$_{.00}$	.21$_{.00}$	16.5$_{0.1}$	**.28**$_{.00}$	**.36**$_{.00}$	.18$_{.00}$	11.2$_{0.1}$	.13$_{.00}$	**.34**$_{.00}$	**.22**$_{.00}$	**12.0**$_{0.1}$	**.11**$_{.00}$
+ CLP+	.32$_{.00}$	**.22**$_{.00}$	**19.2**$_{0.2}$	**.12**$_{.00}$	**.17**$_{.00}$	.20$_{.00}$	**18.6**$_{0.1}$	**.30**$_{.00}$	**.40**$_{.01}$	.18$_{.00}$	11.2$_{0.1}$	.14$_{.00}$	**.38**$_{.01}$	**.22**$_{.00}$	**11.4**$_{0.1}$	**.10**$_{.00}$
BLOOM-7B																
Source	.33$_{.00}$	.22$_{.00}$	6.3$_{0.1}$	.15$_{.01}$	.17$_{.00}$	.20$_{.00}$	8.1$_{0.1}$	.36$_{.00}$	.33$_{.00}$	.18$_{.00}$	2.8$_{0.1}$	.21$_{.00}$	.34$_{.00}$	.22$_{.00}$	7.3$_{0.2}$	.21$_{.00}$
LAPT	.34$_{.00}$	.19$_{.01}$	18.7$_{0.5}$	.17$_{.01}$	.17$_{.00}$	.19$_{.01}$	18.2$_{0.2}$	.36$_{.01}$	.35$_{.00}$	.17$_{.00}$	9.7$_{0.2}$	.22$_{.00}$	.33$_{.00}$	.20$_{.00}$	13.2$_{0.1}$	.14$_{.00}$
+ Heuristics	**.35**$_{.00}$	.20$_{.00}$	17.6$_{0.3}$	**.24**$_{.00}$	**.17**$_{.00}$	**.21**$_{.00}$	16.7$_{0.1}$	**.38**$_{.00}$	.33$_{.00}$	**.19**$_{.00}$	**10.5**$_{0.1}$	.19$_{.00}$	**.36**$_{.00}$	**.23**$_{.00}$	12.3$_{0.1}$	.16$_{.00}$
+ CLP+	**.35**$_{.00}$	.21$_{.00}$	18.0$_{0.3}$	**.22**$_{.00}$	**.17**$_{.00}$	**.20**$_{.00}$	16.9$_{0.1}$	**.37**$_{.01}$	.32$_{.00}$	**.21**$_{.00}$	**10.1**$_{0.1}$	.21$_{.00}$	**.39**$_{.01}$	**.22**$_{.00}$	12.0$_{0.2}$	.18$_{.00}$
TigerBot-7B																
Source	.42$_{.00}$	.22$_{.00}$	5.4$_{0.2}$	.32$_{.03}$	.29$_{.00}$	.28$_{.01}$	1.9$_{0.1}$	.51$_{.01}$	.36$_{.00}$	.20$_{.00}$	2.4$_{0.1}$	.05$_{.00}$	.33$_{.00}$	.22$_{.00}$	9.0$_{0.2}$	.06$_{.00}$
LAPT	.37$_{.00}$	.21$_{.01}$	19.5$_{0.3}$	.20$_{.00}$	.32$_{.01}$	.21$_{.00}$	14.8$_{0.2}$	.50$_{.00}$	.41$_{.00}$	.18$_{.01}$	9.4$_{0.1}$	.12$_{.00}$	.43$_{.00}$	.20$_{.01}$	15.9$_{0.1}$	.15$_{.00}$
+ Heuristics	.35$_{.00}$	.20$_{.01}$	18.8$_{0.2}$	.17$_{.00}$	.17$_{.00}$	.24$_{.00}$	**17.1**$_{0.2}$	.38$_{.00}$	.34$_{.00}$	.19$_{.00}$	6.6$_{0.1}$	.09$_{.00}$	.31$_{.00}$	**.22**$_{.00}$	8.2$_{0.1}$	.04$_{.00}$
+ CLP+	.35$_{.00}$	**.22**$_{.00}$	**20.2**$_{0.1}$	.19$_{.00}$	.17$_{.00}$	.21$_{.00}$	**18.9**$_{0.1}$	.40$_{.00}$	.37$_{.00}$	**.23**$_{.00}$	**9.1**$_{0.2}$	**.16**$_{.01}$	.32$_{.00}$	**.22**$_{.00}$	7.7$_{0.1}$	.05$_{.00}$
Mistral-7B																
Source	.36$_{.00}$	.28$_{.00}$	8.3$_{0.2}$	.35$_{.00}$	.21$_{.00}$	.30$_{.00}$	8.4$_{0.3}$	.56$_{.00}$	.41$_{.00}$	.24$_{.00}$	2.4$_{0.1}$	.22$_{.00}$	.34$_{.00}$	.20$_{.00}$	5.7$_{0.2}$	.12$_{.00}$
LAPT	.35$_{.01}$	.25$_{.01}$	25.2$_{0.3}$	.30$_{.00}$	.19$_{.00}$	.21$_{.01}$	23.1$_{0.1}$	.48$_{.00}$	.34$_{.00}$	.18$_{.00}$	10.9$_{0.1}$	.12$_{.00}$	.36$_{.00}$	.22$_{.01}$	16.5$_{0.1}$	.18$_{.00}$
+ Heuristics	.32$_{.00}$	.22$_{.00}$	20.8$_{0.3}$	.21$_{.00}$	**.27**$_{.00}$	.20$_{.00}$	17.2$_{0.2}$	.37$_{.00}$	.33$_{.00}$	.20$_{.00}$	**11.2**$_{0.2}$	.17$_{.00}$	.30$_{.00}$	**.22**$_{.00}$	11.9$_{0.2}$	.12$_{.00}$
+ CLP+	**.36**$_{.00}$	.22$_{.00}$	20.2$_{0.2}$	.23$_{.00}$	**.29**$_{.00}$	.23$_{.00}$	15.9$_{0.2}$	.38$_{.00}$	.38$_{.00}$	.20$_{.00}$	10.8$_{0.1}$	**.24**$_{.00}$	.31$_{.01}$	.21$_{.00}$	12.6$_{0.1}$	.17$_{.00}$
BLOOM-1B							**Few-shot**									
Source	.35$_{.00}$	.19$_{.00}$	-	.11$_{.00}$	.28$_{.00}$	.18$_{.00}$	-	.24$_{.00}$	.34$_{.00}$	.18$_{.00}$	-	.21$_{.01}$	.35$_{.00}$	.22$_{.00}$	-	.03$_{.00}$
LAPT	.34$_{.01}$	.18$_{.01}$	-	.14$_{.00}$	.30$_{.01}$	.21$_{.01}$	-	.27$_{.00}$	.32$_{.01}$	.17$_{.01}$	-	.18$_{.00}$	.36$_{.00}$	.20$_{.01}$	-	.02$_{.00}$
+ Random	**.35**$_{.00}$	**.21**$_{.00}$	-	**.16**$_{.00}$	.28$_{.00}$	.19$_{.00}$	-	.24$_{.00}$	**.35**$_{.00}$	**.19**$_{.01}$	-	.18$_{.00}$	.34$_{.00}$	.18$_{.00}$	-	**.03**$_{.00}$
+ Heuristics	**.35**$_{.00}$	**.22**$_{.00}$	-	**.18**$_{.01}$	**.34**$_{.00}$	.16$_{.00}$	-	**.29**$_{.00}$	.33$_{.00}$	**.21**$_{.00}$	-	.19$_{.00}$	.33$_{.00}$	.18$_{.00}$	-	**.06**$_{.00}$
+ CLP	**.36**$_{.00}$	**.20**$_{.00}$	-	**.18**$_{.00}$	**.37**$_{.00}$	.19$_{.00}$	-	**.33**$_{.00}$	**.35**$_{.00}$	**.20**$_{.01}$	-	.20$_{.00}$	.35$_{.00}$	.19$_{.00}$	-	**.07**$_{.00}$
+ FOCUS	.34$_{.00}$	**.19**$_{.00}$	-	**.19**$_{.01}$	**.54**$_{.00}$	**.21**$_{.00}$	-	**.33**$_{.00}$	**.34**$_{.00}$	**.19**$_{.01}$	-	.20$_{.00}$	.33$_{.01}$	.20$_{.00}$	-	**.07**$_{.00}$
+ CLP+	.34$_{.00}$	**.19**$_{.00}$	-	**.22**$_{.00}$	**.51**$_{.00}$	.20$_{.00}$	-	**.34**$_{.00}$	**.35**$_{.01}$	**.20**$_{.01}$	-	.19$_{.00}$	.35$_{.01}$	.18$_{.00}$	-	**.07**$_{.00}$
BLOOM-7B																
Source	.42$_{.00}$	.20$_{.01}$	-	.28$_{.00}$	.23$_{.00}$	.18$_{.00}$	-	.33$_{.00}$	.43$_{.00}$	.20$_{.01}$	-	.30$_{.00}$	.38$_{.00}$	.18$_{.00}$	-	.11$_{.00}$
LAPT	.34$_{.01}$	.21$_{.01}$	-	.28$_{.00}$	.34$_{.01}$	.20$_{.01}$	-	.36$_{.01}$	.38$_{.01}$	.18$_{.01}$	-	.28$_{.00}$	.38$_{.01}$	.20$_{.01}$	-	.09$_{.00}$
+ Heuristics	.36$_{.00}$	.20$_{.00}$	-	**.31**$_{.00}$	.26$_{.00}$	**.21**$_{.00}$	-	**.43**$_{.00}$	.37$_{.00}$	**.20**$_{.00}$	-	**.32**$_{.00}$	.33$_{.00}$	**.20**$_{.00}$	-	**.12**$_{.00}$
+ CLP+	.37$_{.00}$	**.22**$_{.00}$	-	**.29**$_{.00}$	**.37**$_{.01}$	**.20**$_{.00}$	-	**.43**$_{.00}$	.32$_{.00}$	**.21**$_{.00}$	-	**.35**$_{.00}$	.35$_{.00}$	**.20**$_{.00}$	-	**.13**$_{.00}$
TigerBot-7B																
Source	.43$_{.00}$	.38$_{.01}$	-	.38$_{.00}$	.42$_{.01}$	.33$_{.00}$	-	.45$_{.00}$	.39$_{.00}$	.18$_{.00}$	-	.13$_{.00}$	.36$_{.00}$	.23$_{.00}$	-	.04$_{.00}$
LAPT	.46$_{.01}$	.39$_{.00}$	-	.37$_{.00}$	.29$_{.00}$	.31$_{.00}$	-	.47$_{.00}$	.43$_{.01}$	.19$_{.01}$	-	.20$_{.01}$	.44$_{.01}$	.21$_{.00}$	-	.15$_{.00}$
+ Heuristics	.35$_{.00}$	.27$_{.00}$	-	.24$_{.01}$	**.47**$_{.00}$	.24$_{.00}$	-	.42$_{.00}$	.33$_{.00}$	**.20**$_{.00}$	-	.11$_{.00}$	.36$_{.00}$	.22$_{.00}$	-	.02$_{.00}$
+ CLP+	.36$_{.00}$	.32$_{.00}$	-	.34$_{.00}$	.32$_{.00}$	.21$_{.00}$	-	.43$_{.00}$	.35$_{.00}$	**.20**$_{.00}$	-	**.25**$_{.00}$	.34$_{.00}$	.19$_{.00}$	-	.04$_{.00}$
Mistral-7B																
Source	.54$_{.00}$	.55$_{.00}$	-	.45$_{.00}$	.49$_{.00}$	.42$_{.00}$	-	.56$_{.00}$	.46$_{.00}$	.35$_{.00}$	-	.32$_{.00}$	.38$_{.00}$	.21$_{.00}$	-	.20$_{.00}$
LAPT	.51$_{.01}$	.47$_{.00}$	-	.28$_{.00}$	.43$_{.01}$	.37$_{.01}$	-	.55$_{.00}$	.44$_{.01}$	.30$_{.01}$	-	.22$_{.00}$	.47$_{.01}$	.34$_{.01}$	-	.25$_{.00}$
+ Heuristics	.40$_{.00}$	.40$_{.00}$	-	.24$_{.00}$	.39$_{.00}$	.24$_{.00}$	-	.44$_{.00}$	.35$_{.00}$	.18$_{.00}$	-	.24$_{.00}$	.33$_{.00}$	.17$_{.00}$	-	.13$_{.00}$
+ CLP+	.37$_{.00}$	.48$_{.00}$	-	.30$_{.00}$	.31$_{.00}$	.26$_{.00}$	-	.44$_{.00}$	.34$_{.00}$	.23$_{.00}$	-	**.33**$_{.00}$	.32$_{.00}$	.19$_{.00}$	-	.14$_{.00}$

Table 9: Mean performances over five runs with standard deviations when prompted in English on 500 randomly selected test samples for each dataset. The baselines are in grey. **Bold** indicates comparable or better results than the baselines.

Approach	German				Japanese				Arabic				Swahili			
	NLI	MC	SUM	SPAN	NLI	MC	SUM	SPAN	NLI	MC	SUM	SPAN	NLI	MC	SUM	SPAN
BLOOM-1B							<u>**Zero-shot**</u>									
Source	.33$_{.00}$	.21$_{.00}$	17.8$_{0.3}$	.06$_{.00}$	.29$_{.00}$	.20$_{.00}$	18.2$_{0.3}$	.22$_{.00}$	.35$_{.00}$	.20$_{.00}$	12.0$_{0.2}$	.15$_{.01}$	.32$_{.00}$	.22$_{.00}$	12.0$_{0.3}$	.03$_{.00}$
LAPT	.36$_{.01}$	.22$_{.01}$	14.3$_{0.2}$	.09$_{.00}$	.28$_{.00}$	.20$_{.00}$	20.7$_{0.2}$	.26$_{.00}$	.36$_{.00}$	.19$_{.01}$	11.4$_{0.1}$	.13$_{.01}$	.31$_{.01}$	.18$_{.00}$	7.7$_{0.1}$	.07$_{.00}$
+ Random	.35$_{.00}$	**.22**$_{.00}$	15.3$_{0.2}$	**.14**$_{.00}$	**.29**$_{.00}$	**.21**$_{.00}$	19.0$_{0.0}$	**.32**$_{.00}$	.35$_{.00}$	.19$_{.00}$	11.5$_{0.0}$	.14$_{.01}$	**.33**$_{.00}$	**.22**$_{.00}$	10.2$_{0.1}$	**.08**$_{.01}$
+ Heuristics	.34$_{.00}$	.19$_{.00}$	15.3$_{0.2}$	**.13**$_{.00}$	**.29**$_{.00}$	.19$_{.00}$	19.2$_{0.0}$	**.31**$_{.00}$	**.36**$_{.00}$	**.22**$_{.02}$	11.3$_{0.1}$	.13$_{.00}$	**.34**$_{.00}$	**.22**$_{.00}$	11.9$_{0.2}$	**.11**$_{.00}$
+ CLP	.34$_{.00}$	.18$_{.00}$	14.6$_{0.7}$	**.14**$_{.00}$	**.29**$_{.00}$	**.25**$_{.00}$	18.8$_{0.1}$	**.33**$_{.00}$	**.36**$_{.00}$	**.21**$_{.00}$	11.2$_{0.2}$	.14$_{.00}$	**.34**$_{.00}$	**.22**$_{.00}$	11.5$_{0.2}$	**.11**$_{.00}$
+ FOCUS	.34$_{.00}$	.19$_{.00}$	16.1$_{0.8}$	**.13**$_{.01}$	**.29**$_{.00}$	**.21**$_{.00}$	19.2$_{0.0}$	**.33**$_{.00}$	.35$_{.00}$	**.20**$_{.01}$	11.2$_{0.1}$	.14$_{.00}$	**.34**$_{.00}$	**.22**$_{.00}$	11.2$_{0.1}$	**.12**$_{.00}$
+ CLP+	.31$_{.00}$	.15$_{.00}$	15.8$_{0.6}$	**.13**$_{.00}$	**.29**$_{.00}$	.19$_{.00}$	19.4$_{0.0}$	**.33**$_{.00}$	.35$_{.00}$	.17$_{.00}$	11.3$_{0.1}$	**.15**$_{.01}$	**.34**$_{.00}$	.20$_{.00}$	10.4$_{0.1}$	**.10**$_{.00}$
BLOOM-7B																
Source	.36$_{.00}$	.21$_{.00}$	23.1$_{0.2}$	.15$_{.01}$	.28$_{.00}$	.21$_{.00}$	19.0$_{0.2}$	.33$_{.00}$	.38$_{.00}$	.17$_{.00}$	11.5$_{0.1}$	.25$_{.00}$	.36$_{.01}$	.22$_{.00}$	14.3$_{0.1}$	.22$_{.01}$
LAPT	.37$_{.01}$	.21$_{.01}$	19.4$_{0.3}$	.14$_{.00}$	.21$_{.00}$	.21$_{.00}$	21.6$_{0.1}$	.36$_{.00}$	.36$_{.00}$	.16$_{.00}$	11.5$_{0.2}$	.21$_{.00}$	.36$_{.01}$	.20$_{.00}$	13.0$_{0.1}$	.14$_{.00}$
+ Heuristics	.32$_{.00}$	**.22**$_{.00}$	19.7$_{0.3}$	**.21**$_{.00}$	**.30**$_{.00}$	**.23**$_{.00}$	19.5$_{0.1}$	**.38**$_{.00}$	.37$_{.00}$	**.19**$_{.00}$	10.7$_{0.2}$	.21$_{.00}$	.33$_{.00}$	**.22**$_{.00}$	11.6$_{0.1}$	.16$_{.00}$
+ CLP+	.35$_{.00}$	**.21**$_{.00}$	18.7$_{0.7}$	**.20**$_{.01}$	**.29**$_{.00}$	**.21**$_{.00}$	19.5$_{0.1}$	**.40**$_{.00}$	**.38**$_{.00}$	**.21**$_{.00}$	11.0$_{0.1}$	.21$_{.00}$	.33$_{.00}$	**.23**$_{.00}$	10.9$_{0.1}$	.17$_{.00}$
TigerBot-7B																
Source	.33$_{.00}$	.24$_{.00}$	23.9$_{0.2}$	.26$_{.01}$	.17$_{.00}$	.24$_{.00}$	19.4$_{0.3}$	.57$_{.01}$	.33$_{.00}$	.21$_{.00}$	9.0$_{0.1}$	.04$_{.00}$	.29$_{.01}$	.22$_{.00}$	12.4$_{0.2}$	.03$_{.00}$
LAPT	.32$_{.00}$	.21$_{.00}$	18.5$_{0.2}$	.18$_{.00}$	.17$_{.00}$	.21$_{.01}$	21.6$_{0.1}$	.49$_{.01}$	.33$_{.00}$	.18$_{.00}$	9.8$_{0.2}$	.13$_{.00}$	.31$_{.00}$	.21$_{.01}$	15.9$_{0.1}$	.10$_{.00}$
+ Heuristics	**.35**$_{.00}$	.20$_{.00}$	16.1$_{0.3}$	.18$_{.00}$	**.29**$_{.00}$	.22$_{.00}$	19.6$_{0.1}$	.40$_{.00}$	**.36**$_{.00}$	.17$_{.00}$	**10.3**$_{0.3}$	.08$_{.00}$	**.36**$_{.00}$	**.22**$_{.00}$	8.1$_{0.1}$	.05$_{.00}$
+ CLP+	**.33**$_{.00}$	.20$_{.01}$	14.1$_{0.3}$	.19$_{.00}$	**.29**$_{.00}$	.20$_{.00}$	19.8$_{0.0}$	.41$_{.00}$	**.38**$_{.00}$	**.22**$_{.00}$	**11.2**$_{0.3}$	**.16**$_{.00}$	.30$_{.01}$	**.22**$_{.00}$	8.6$_{0.1}$	.09$_{.00}$
Mistral-7B																
Source	.34$_{.00}$	.25$_{.00}$	24.1$_{0.2}$	.35$_{.01}$	.17$_{.00}$	.28$_{.00}$	23.7$_{0.1}$	.60$_{.00}$	.33$_{.00}$	.20$_{.00}$	11.2$_{0.1}$	.21$_{.00}$	.35$_{.00}$	.22$_{.00}$	15.4$_{0.1}$	.07$_{.00}$
LAPT	.33$_{.01}$	.25$_{.02}$	24.2$_{0.2}$	.28$_{.01}$	.17$_{.00}$	.20$_{.01}$	23.4$_{0.1}$	.60$_{.00}$	.33$_{.00}$	.18$_{.00}$	10.8$_{0.1}$	.14$_{.01}$	.33$_{.01}$	.22$_{.01}$	16.2$_{0.2}$	.12$_{.00}$
+ Heuristics	**.40**$_{.00}$	**.26**$_{.00}$	21.2$_{0.1}$	.22$_{.00}$	**.29**$_{.00}$	.20$_{.00}$	19.7$_{0.1}$	.43$_{.00}$	**.39**$_{.00}$	.19$_{.00}$	10.7$_{0.1}$	.13$_{.00}$	.34$_{.00}$	**.22**$_{.00}$	10.6$_{0.1}$	**.14**$_{.00}$
+ CLP+	**.35**$_{.00}$	**.25**$_{.00}$	20.2$_{0.3}$	.21$_{.00}$	**.28**$_{.00}$	.20$_{.00}$	19.9$_{0.0}$	.46$_{.00}$	**.38**$_{.00}$	.16$_{.00}$	**11.5**$_{0.2}$	**.21**$_{.00}$	.33$_{.00}$	.21$_{.00}$	10.2$_{0.1}$	**.16**$_{.00}$
BLOOM-1B							<u>**Few-shot**</u>									
Source	.38$_{.00}$	.20$_{.00}$	-	.10$_{.00}$	.44$_{.00}$	.19$_{.00}$	-	.32$_{.00}$	.35$_{.00}$	.17$_{.00}$	-	.20$_{.01}$	.37$_{.00}$	.23$_{.00}$	-	.02$_{.00}$
LAPT	.37$_{.01}$	.17$_{.01}$	-	.13$_{.01}$	.26$_{.01}$	.21$_{.01}$	-	.34$_{.01}$	.34$_{.00}$	.16$_{.00}$	-	.16$_{.00}$	.34$_{.01}$	.19$_{.01}$	-	.02$_{.00}$
+ Random	.34$_{.00}$	**.21**$_{.00}$	-	**.16**$_{.00}$	.29$_{.00}$	**.21**$_{.00}$	-	**.34**$_{.00}$	**.35**$_{.00}$	**.22**$_{.00}$	-	.16$_{.01}$	.34$_{.00}$	.20$_{.00}$	-	**.06**$_{.00}$
+ Heuristics	.33$_{.00}$	**.23**$_{.00}$	-	**.17**$_{.00}$	.30$_{.00}$	**.22**$_{.00}$	-	.32$_{.00}$	**.36**$_{.00}$	**.21**$_{.01}$	-	.15$_{.01}$	.33$_{.00}$	.19$_{.00}$	-	**.07**$_{.00}$
+ CLP	.34$_{.00}$	**.21**$_{.00}$	-	**.17**$_{.01}$	.30$_{.00}$	.20$_{.00}$	-	.33$_{.00}$	**.35**$_{.00}$	**.21**$_{.00}$	-	.15$_{.01}$	.33$_{.00}$	.19$_{.00}$	-	**.08**$_{.00}$
+ FOCUS	.34$_{.00}$	.18$_{.00}$	-	**.17**$_{.01}$	.27$_{.00}$	.20$_{.00}$	-	**.36**$_{.00}$	**.36**$_{.00}$	**.20**$_{.00}$	-	.15$_{.00}$	.34$_{.00}$	.19$_{.00}$	-	**.08**$_{.00}$
+ CLP+	.34$_{.00}$	**.20**$_{.00}$	-	**.19**$_{.00}$	.29$_{.00}$	**.22**$_{.00}$	-	**.36**$_{.00}$	**.37**$_{.01}$	**.20**$_{.01}$	-	.15$_{.01}$	.30$_{.00}$	.18$_{.00}$	-	**.08**$_{.00}$
BLOOM-7B																
Source	.38$_{.00}$	.23$_{.00}$	-	.29$_{.00}$	.41$_{.01}$	.19$_{.00}$	-	.49$_{.01}$	.37$_{.00}$	.18$_{.00}$	-	.29$_{.00}$	.34$_{.00}$	.18$_{.00}$	-	.11$_{.00}$
LAPT	.35$_{.00}$	.24$_{.00}$	-	.23$_{.00}$	.34$_{.01}$	.19$_{.01}$	-	.53$_{.01}$	.36$_{.01}$	.18$_{.01}$	-	.23$_{.00}$	.37$_{.00}$	.18$_{.01}$	-	.07$_{.00}$
+ Heuristics	.33$_{.00}$	.22$_{.00}$	-	.28$_{.00}$	.28$_{.00}$	**.21**$_{.00}$	-	.46$_{.00}$	.36$_{.00}$	**.21**$_{.00}$	-	.24$_{.00}$	.36$_{.00}$	**.19**$_{.00}$	-	**.13**$_{.00}$
+ CLP+	.34$_{.00}$	.22$_{.00}$	-	.25$_{.00}$	.30$_{.00}$	**.20**$_{.00}$	-	.46$_{.00}$	.36$_{.00}$	**.22**$_{.00}$	-	.25$_{.00}$	.36$_{.00}$	**.18**$_{.00}$	-	**.13**$_{.00}$
TigerBot-7B																
Source	.31$_{.00}$	.37$_{.00}$	-	.42$_{.00}$	.16$_{.00}$	.34$_{.00}$	-	.65$_{.00}$	.30$_{.00}$	.19$_{.00}$	-	.10$_{.00}$	.36$_{.00}$	.19$_{.01}$	-	.03$_{.00}$
LAPT	.30$_{.00}$	.39$_{.01}$	-	.36$_{.02}$	.16$_{.00}$	.34$_{.01}$	-	.66$_{.00}$	.30$_{.00}$	.20$_{.01}$	-	.17$_{.00}$	.36$_{.00}$	.20$_{.00}$	-	.09$_{.00}$
+ Heuristics	**.33**$_{.00}$	.26$_{.01}$	-	.21$_{.00}$	**.29**$_{.00}$	.24$_{.00}$	-	.49$_{.00}$	**.35**$_{.00}$	**.20**$_{.00}$	-	.09$_{.00}$	.35$_{.00}$	**.21**$_{.00}$	-	.04$_{.00}$
+ CLP+	**.36**$_{.01}$	.31$_{.00}$	-	.31$_{.00}$	**.30**$_{.00}$	.21$_{.00}$	-	.50$_{.00}$	**.37**$_{.01}$	.19$_{.00}$	-	**.19**$_{.00}$	.34$_{.00}$	.18$_{.00}$	-	.06$_{.00}$
Mistral-7B																
Source	.33$_{.00}$	.53$_{.00}$	-	.48$_{.00}$	.16$_{.00}$	.42$_{.00}$	-	.69$_{.00}$	.30$_{.00}$	.32$_{.00}$	-	.31$_{.00}$	.40$_{.00}$	.21$_{.00}$	-	.12$_{.00}$
LAPT	.33$_{.00}$	.46$_{.00}$	-	.27$_{.00}$	.16$_{.00}$	.37$_{.01}$	-	.68$_{.00}$	.30$_{.00}$	.30$_{.01}$	-	.26$_{.00}$	.36$_{.01}$	.34$_{.00}$	-	.21$_{.00}$
+ Heuristics	**.45**$_{.00}$	.41$_{.00}$	-	.24$_{.00}$	**.30**$_{.00}$	.24$_{.00}$	-	.49$_{.00}$	**.34**$_{.00}$	.18$_{.00}$	-	.17$_{.00}$	.34$_{.00}$	.18$_{.00}$	-	.09$_{.00}$
+ CLP+	**.37**$_{.00}$	.47$_{.00}$	-	.25$_{.00}$	**.29**$_{.00}$	.25$_{.00}$	-	.50$_{.00}$	**.38**$_{.00}$	.23$_{.00}$	-	.23$_{.00}$	.33$_{.00}$	.20$_{.00}$	-	.14$_{.00}$

Table 10: Mean performances over five runs with standard deviations when prompted in a target language on 500 randomly selected test samples for each dataset. The baselines are in grey. **Bold** indicates comparable or better results than the baselines.

Approach	NLI				MC				SUM				SPAN			
	de	ja	ar	sw	de	ja	ar	sw	de	ja	ar	sw	de	ja	ar	sw
Zero-shot																
BLOOM-1B																
Source	$.34_{.00}$				$.18_{.00}$				$11.2_{0.1}$				$.21_{.01}$			
LAPT	$.35_{.01}$	$.33_{.00}$	$.33_{.00}$	$.33_{.00}$	$.21_{.01}$	$.20_{.00}$	$.17_{.01}$	$.18_{.01}$	$9.7_{0.1}$	$10.4_{0.0}$	$10.6_{0.1}$	$10.1_{0.0}$	$.15_{.01}$	$.18_{.00}$	$.17_{.00}$	$.15_{.01}$
+ Heuristics	$\mathbf{.35}_{.00}$	$\mathbf{.34}_{.00}$	$\mathbf{.35}_{.00}$	$\mathbf{.36}_{.00}$	$.20_{.00}$	$\mathbf{.20}_{.00}$	$\mathbf{.21}_{.00}$	$\mathbf{.20}_{.00}$	$\mathbf{11.2}_{0.1}$	$\mathbf{11.7}_{0.1}$	$\mathbf{12.3}_{0.1}$	$10.1_{0.2}$	$.10_{.01}$	$.06_{.01}$	$.07_{.00}$	$.09_{.01}$
+ CLP+	$\mathbf{.35}_{.00}$	$\mathbf{.34}_{.00}$	$.32_{.00}$	$\mathbf{.37}_{.01}$	$.19_{.00}$	$\mathbf{.20}_{.00}$	$\mathbf{.20}_{.00}$	$\mathbf{.20}_{.00}$	$10.8_{0.1}$	$\mathbf{12.1}_{0.1}$	$\mathbf{12.4}_{0.1}$	$10.4_{0.2}$	$.12_{.00}$	$.11_{.00}$	$.07_{.00}$	$.08_{.00}$
BLOOM-7B																
Source	$.36_{.00}$				$.17_{.00}$				$11.1_{0.1}$				$.31_{.00}$			
LAPT	$.34_{.00}$	$.34_{.00}$	$.34_{.00}$	$.36_{.00}$	$.20_{.01}$	$.20_{.01}$	$.20_{.01}$	$.18_{.01}$	$10.8_{0.0}$	$11.0_{0.0}$	$10.9_{0.0}$	$10.6_{0.1}$	$.25_{.00}$	$.27_{.01}$	$.28_{.00}$	$.23_{.00}$
+ Heuristics	$\mathbf{.36}_{.00}$	$.34_{.00}$	$.33_{.00}$	$\mathbf{.36}_{.00}$	$\mathbf{.20}_{.00}$	$\mathbf{.20}_{.00}$	$\mathbf{.20}_{.00}$	$\mathbf{.19}_{.00}$	$\mathbf{11.2}_{0.1}$	$\mathbf{11.1}_{0.1}$	$\mathbf{12.2}_{0.1}$	$10.5_{0.0}$	$.24_{.01}$	$.19_{.00}$	$.17_{.00}$	$.17_{.00}$
+ CLP+	$\mathbf{.36}_{.00}$	$.34_{.00}$	$.33_{.00}$	$\mathbf{.38}_{.00}$	$\mathbf{.20}_{.00}$	$\mathbf{.20}_{.00}$	$\mathbf{.20}_{.00}$	$\mathbf{.20}_{.00}$	$10.5_{0.1}$	$\mathbf{11.8}_{0.1}$	$\mathbf{12.4}_{0.0}$	$11.1_{0.1}$	$.21_{.00}$	$.10_{.01}$	$.22_{.00}$	$.19_{.00}$
TigerBot-7B																
Source	$.48_{.00}$				$.29_{.00}$				$12.7_{0.1}$				$.42_{.01}$			
LAPT	$.39_{.00}$	$.49_{.01}$	$.45_{.01}$	$.45_{.01}$	$.23_{.00}$	$.25_{.00}$	$.25_{.00}$	$.28_{.01}$	$11.6_{0.1}$	$11.9_{0.1}$	$12.2_{0.1}$	$11.0_{0.1}$	$.31_{.00}$	$.34_{.01}$	$.27_{.00}$	$.35_{.01}$
+ Heuristics	$.37_{.00}$	$.34_{.00}$	$.35_{.00}$	$.31_{.00}$	$.21_{.00}$	$.24_{.00}$	$.20_{.00}$	$.20_{.00}$	$10.2_{0.1}$	$12.1_{0.1}$	$9.8_{0.1}$	$4.8_{0.1}$	$.15_{.00}$	$.24_{.01}$	$.03_{.00}$	$.02_{.00}$
+ CLP+	$.39_{.00}$	$.36_{.00}$	$.36_{.00}$	$.31_{.00}$	$.24_{.00}$	$.24_{.00}$	$.21_{.00}$	$.20_{.00}$	$10.4_{0.1}$	$12.5_{0.1}$	$11.3_{0.1}$	$6.3_{0.1}$	$.20_{.00}$	$.30_{.00}$	$.20_{.00}$	$.03_{.00}$
Mistral-7B																
Source	$.42_{.00}$				$.46_{.00}$				$12.4_{0.2}$				$.44_{.00}$			
LAPT	$.36_{.01}$	$.49_{.01}$	$.45_{.01}$	$.42_{.00}$	$.34_{.01}$	$.32_{.01}$	$.28_{.01}$	$.38_{.01}$	$11.6_{0.0}$	$11.3_{0.1}$	$8.1_{0.2}$	$10.6_{0.1}$	$.39_{.00}$	$.40_{.01}$	$.28_{.00}$	$.36_{.00}$
+ Heuristics	$.33_{.00}$	$.39_{.00}$	$.36_{.00}$	$.33_{.00}$	$.21_{.00}$	$.21_{.00}$	$.20_{.00}$	$.20_{.00}$	$\mathbf{12.5}_{0.1}$	$\mathbf{12.9}_{0.1}$	$11.5_{0.2}$	$8.5_{0.2}$	$.23_{.00}$	$.30_{.00}$	$.19_{.00}$	$.06_{.00}$
+ CLP+	$.36_{.00}$	$.37_{.00}$	$.34_{.00}$	$.31_{.01}$	$.21_{.00}$	$.25_{.00}$	$.19_{.00}$	$.22_{.00}$	$12.2_{0.1}$	$\mathbf{13.1}_{0.1}$	$\mathbf{12.9}_{0.1}$	$10.6_{0.1}$	$.26_{.00}$	$.27_{.01}$	$.31_{.00}$	$.18_{.00}$
Few-shot																
BLOOM-1B																
Source	$.33_{.00}$				$.20_{.00}$				-				$.28_{.00}$			
LAPT	$.34_{.01}$	$.31_{.00}$	$.32_{.01}$	$.33_{.00}$	$.17_{.01}$	$.18_{.01}$	$.19_{.01}$	$.20_{.01}$	-	-	-	-	$.20_{.01}$	$.23_{.01}$	$.25_{.00}$	$.20_{.01}$
LoRA + Heuristics	$.33_{.00}$	$\mathbf{.34}_{.00}$	$.32_{.00}$	$\mathbf{.34}_{.00}$	$.19_{.00}$	$\mathbf{.20}_{.00}$	$\mathbf{.21}_{.00}$	$.19_{.00}$	-	-	-	-	$.16_{.00}$	$.11_{.00}$	$.14_{.00}$	$.17_{.01}$
LoRA + CLP+	$\mathbf{.37}_{.00}$	$\mathbf{.34}_{.00}$	$\mathbf{.35}_{.00}$	$\mathbf{.35}_{.00}$	$.19_{.00}$	$\mathbf{.21}_{.00}$	$\mathbf{.22}_{.00}$	$\mathbf{.20}_{.00}$	-	-	-	-	$.17_{.00}$	$.10_{.01}$	$.13_{.00}$	$.18_{.00}$
BLOOM-7B																
Source	$.43_{.00}$				$.21_{.00}$				-				$.39_{.00}$			
LAPT	$.36_{.01}$	$.38_{.01}$	$.40_{.00}$	$.39_{.01}$	$.20_{.01}$	$.21_{.01}$	$.21_{.00}$	$.19_{.00}$	-	-	-	-	$.36_{.00}$	$.38_{.00}$	$.38_{.00}$	$.37_{.00}$
+ Heuristics	$.36_{.00}$	$.35_{.00}$	$.31_{.00}$	$.32_{.00}$	$.20_{.00}$	$.19_{.00}$	$\mathbf{.22}_{.00}$	$\mathbf{.21}_{.00}$	-	-	-	-	$.37_{.00}$	$.33_{.00}$	$.36_{.00}$	$.34_{.00}$
+ CLP+	$.36_{.00}$	$.33_{.00}$	$.33_{.00}$	$.34_{.00}$	$.18_{.00}$	$\mathbf{.21}_{.00}$	$.20_{.00}$	$.20_{.00}$	-	-	-	-	$.35_{.00}$	$.34_{.00}$	$.36_{.03}$	$.36_{.00}$
TigerBot-7B																
Source	$.49_{.00}$				$.58_{.00}$				-				$.47_{.00}$			
LAPT	$.47_{.01}$	$.56_{.01}$	$.45_{.00}$	$.56_{.01}$	$.57_{.00}$	$.58_{.00}$	$.52_{.00}$	$.52_{.01}$	-	-	-	-	$.47_{.00}$	$.46_{.00}$	$.44_{.00}$	$.47_{.00}$
+ Heuristics	$.36_{.00}$	$.35_{.00}$	$.37_{.00}$	$.34_{.00}$	$.20_{.00}$	$.31_{.00}$	$.23_{.00}$	$.19_{.00}$	-	-	-	-	$.24_{.00}$	$.41_{.01}$	$.03_{.00}$	$.01_{.00}$
+ CLP+	$.42_{.00}$	$.34_{.00}$	$.37_{.00}$	$.36_{.00}$	$.32_{.00}$	$.24_{.00}$	$.19_{.00}$	$.19_{.00}$	-	-	-	-	$.37_{.00}$	$.43_{.00}$	$.29_{.00}$	$.02_{.00}$
Mistral-7B																
Source	$.60_{.00}$				$.66_{.00}$				-				$.51_{.00}$			
LAPT	$.55_{.00}$	$.53_{.01}$	$.49_{.01}$	$.56_{.01}$	$.62_{.00}$	$.59_{.01}$	$.57_{.01}$	$.63_{.01}$	-	-	-	-	$.38_{.00}$	$.49_{.00}$	$.31_{.00}$	$.49_{.00}$
+ Heuristics	$.43_{.00}$	$.35_{.00}$	$.36_{.00}$	$.33_{.00}$	$.33_{.00}$	$.31_{.00}$	$.20_{.00}$	$.19_{.00}$	-	-	-	-	$.27_{.00}$	$.47_{.00}$	$.32_{.00}$	$.05_{.00}$
+ CLP+	$.38_{.00}$	$.34_{.00}$	$.38_{.00}$	$.37_{.00}$	$.45_{.00}$	$.31_{.00}$	$.26_{.00}$	$.21_{.00}$	-	-	-	-	$.34_{.00}$	$.45_{.00}$	$.44_{.00}$	$.28_{.01}$

Table 11: Mean performances over five runs with standard deviations on 500 randomly selected test samples for each English dataset. The baselines are in grey. **Bold** indicates comparable or better results than the baselines.

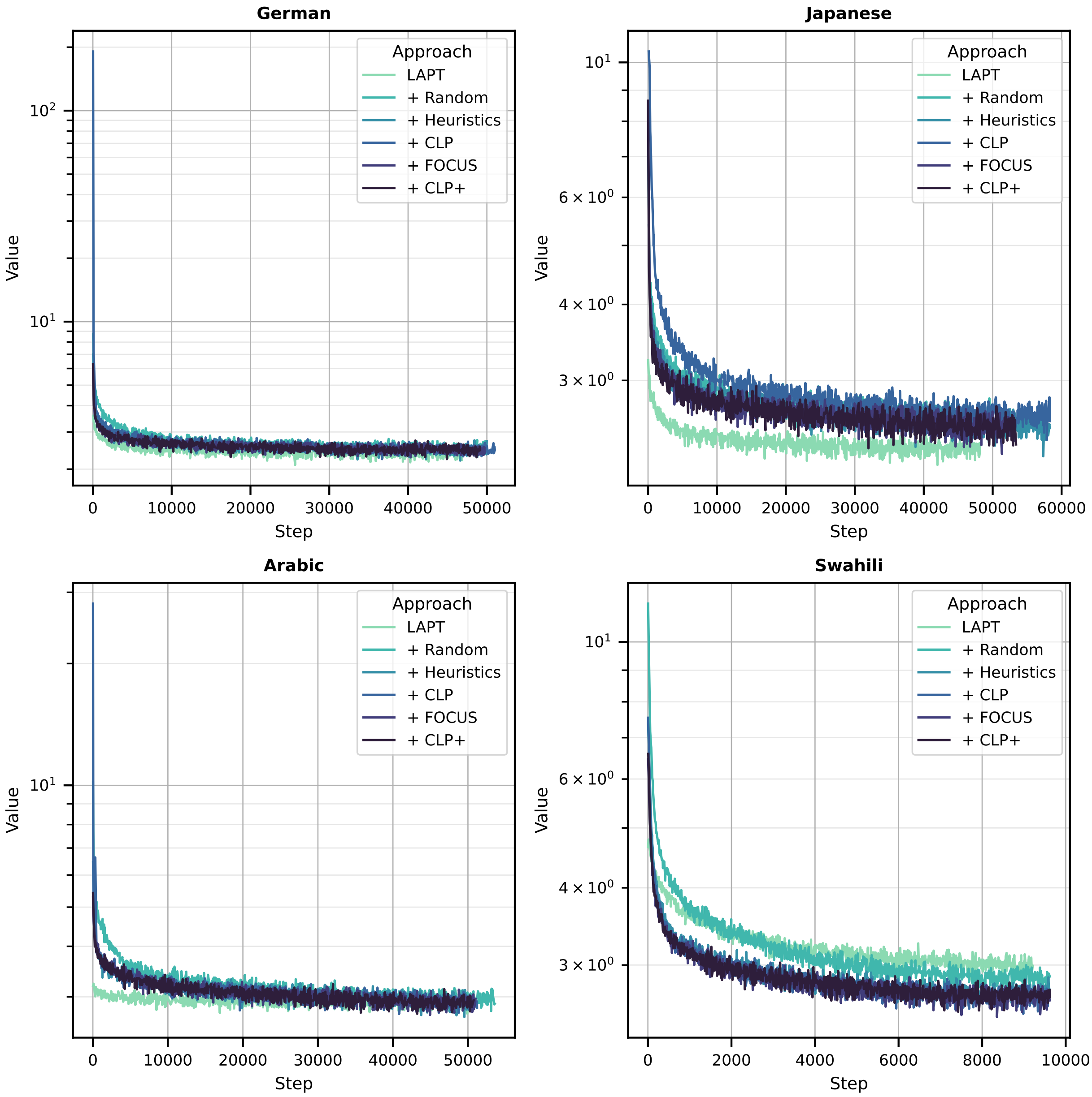

Figure 7: LAPT loss curves for BLOOM-1B

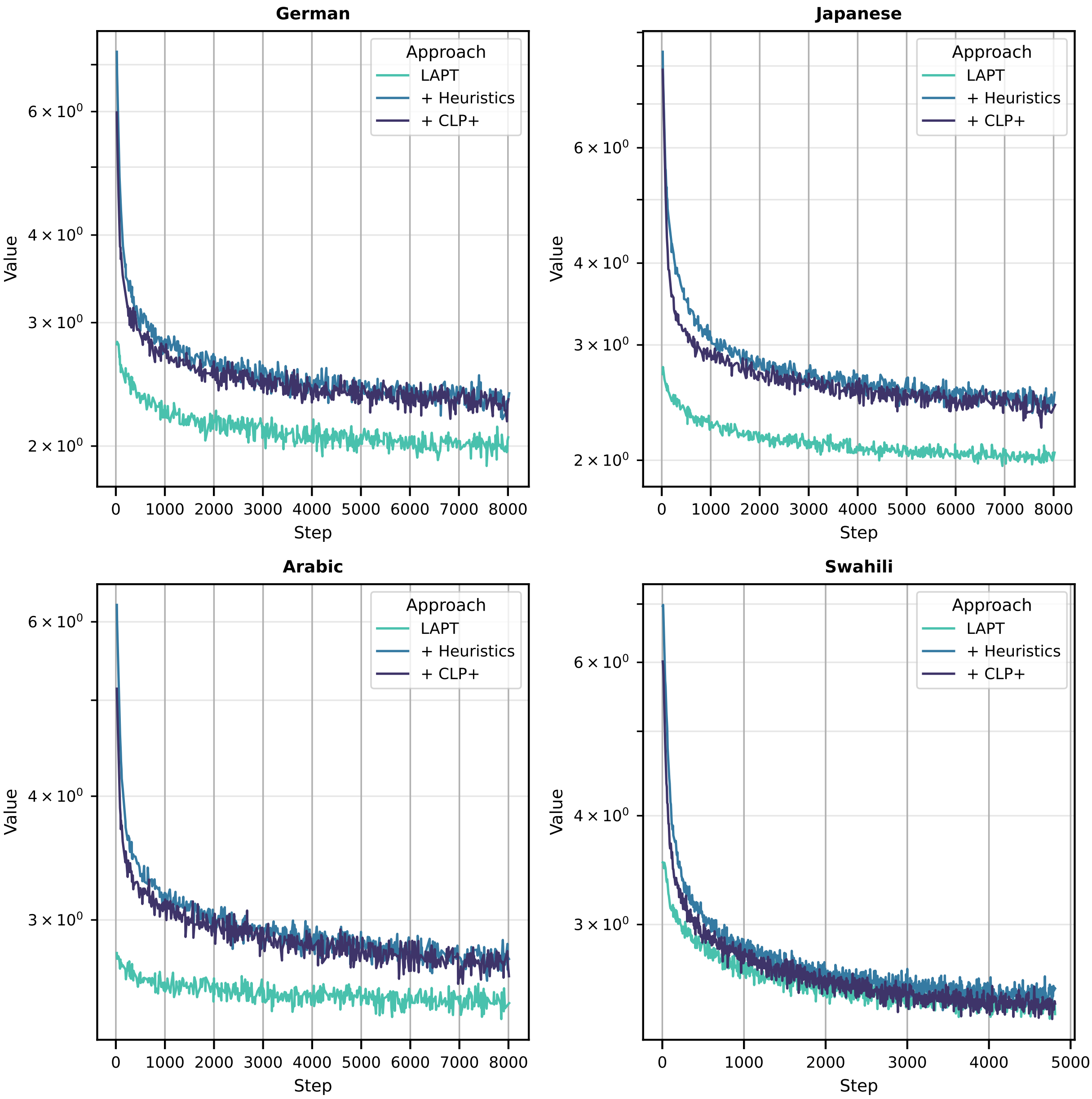

Figure 8: LAPT loss curves for BLOOM-7B

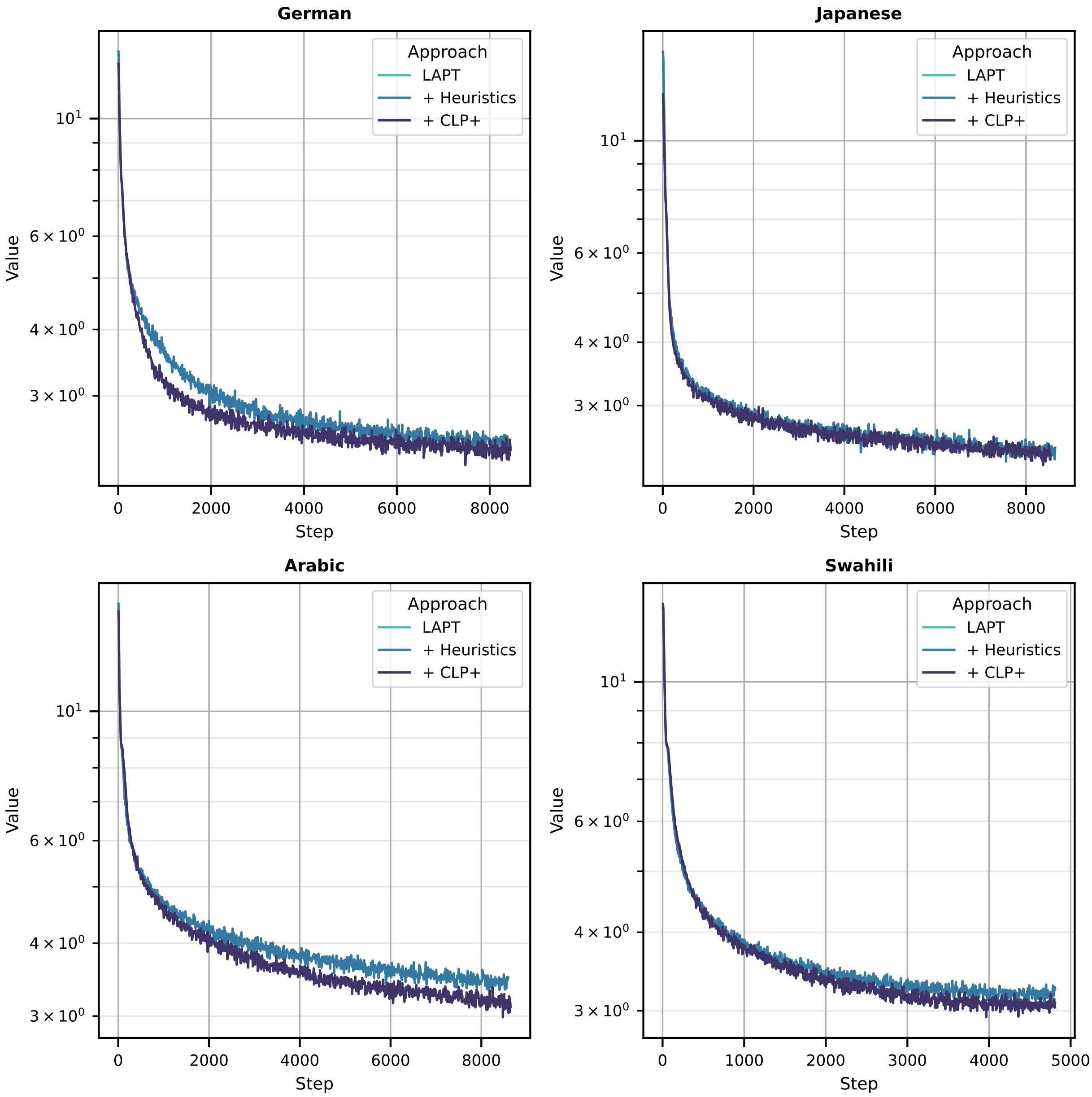

Figure 9: LAPT loss curves for TigerBot-7B

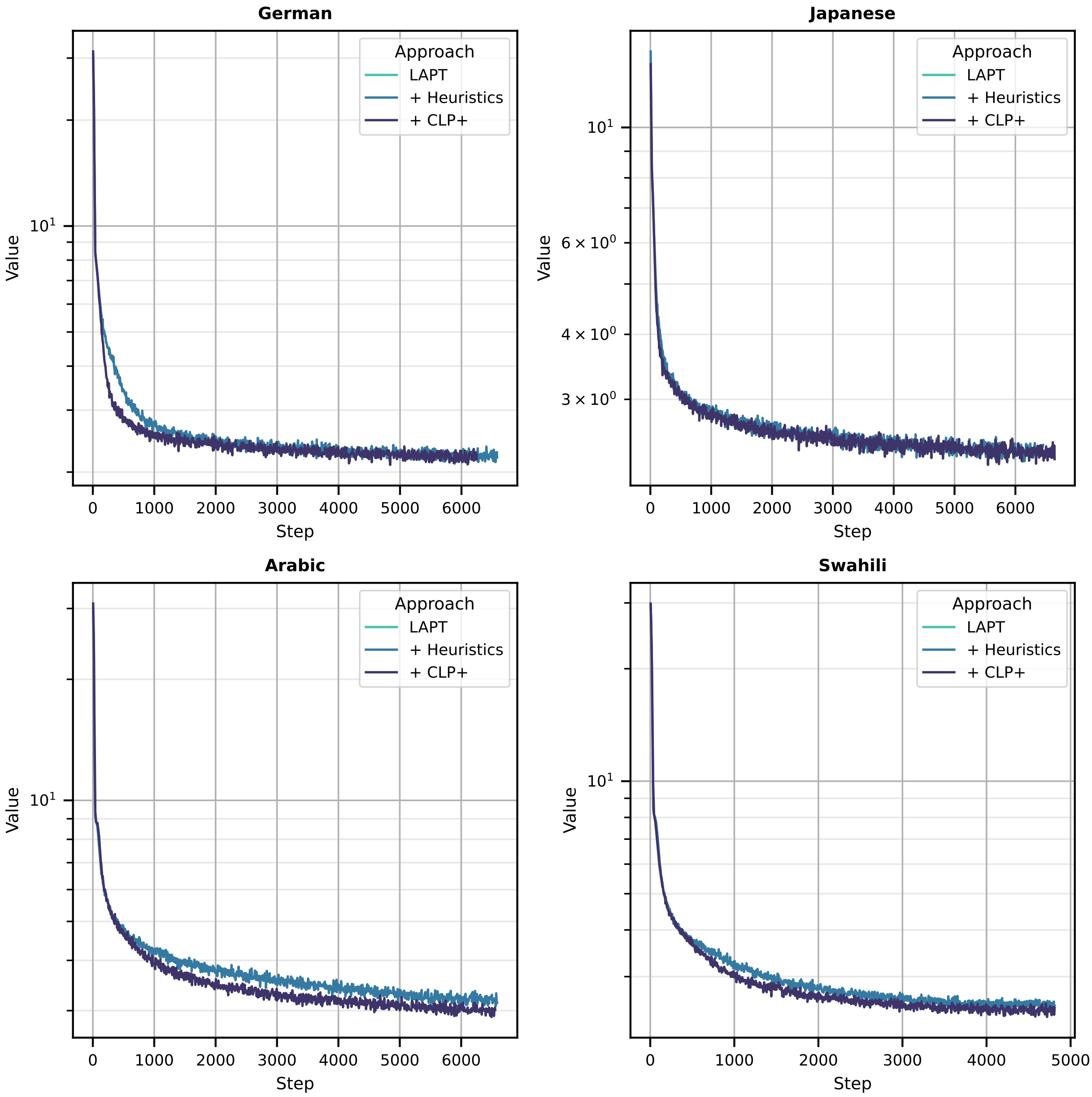

Figure 10: LAPT loss curves for Mistral-7B