Exploring Design Choices for Building Language-Specific LLMs

Atula Tejaswi*, **Nilesh Gupta***, **Eunsol Choi**
Department of Computer Science
The University of Texas at Austin
{atutej, nileshgupta2797, eunsol}@utexas.edu

Abstract

Despite rapid progress in large language models (LLMs), their performance on a vast majority of languages remains unsatisfactory. In this paper, we study building language-specific LLMs by adapting monolingual and multilingual LLMs. We conduct systematic experiments on how design choices (base model selection, vocabulary extension, and continued pretraining) impact the adapted LLM, both in terms of efficiency (how many tokens are needed to encode the same amount of information) and end task performance. We find that (1) the initial performance of LLM does not always correlate with the final performance after the adaptation. Adapting an English-centric models can yield better results than adapting multilingual models despite their worse initial performance on low-resource languages. (2) Efficiency can easily improved with simple vocabulary extension and continued pretraining in most LLMs we study, and (3) The optimal adaptation method (choice of the base model, new vocabulary size, training data, initialization strategy) is highly language-dependent, and the simplest embedding initialization works well across various experimental settings. Together, our work lays foundations on efficiently building language-specific LLMs by adapting existing LLMs.

1 Introduction

The predominance of English data on the internet, combined with a financial interest in English-centric applications, has led to the development of primarily monolingual LLMs (Touvron et al., 2023; Jiang et al., 2023; Abdin et al., 2024b) which exhibit significantly higher proficiency in English compared to other languages. Even when LLMs support other languages, their performance lags behind – both in terms of end task performance and efficiency, measured by the amount of tokens required to encode information (Ahia et al., 2023).

Prior work has mainly focused on building multilingual models that cover a broad spectrum of languages (Scao et al., 2023; Üstün et al., 2024; Lin et al., 2021), or building language-specific LLMs from scratch (Zeng et al., 2023a; Müller and Laurent, 2022; Sengupta et al., 2023). Training a new language-specific LLM from scratch can be expensive, both in terms of compute and data requirements. Therefore, recent efforts have focused on adapting existing, high-performing LLMs (Cui et al., 2023; Levine et al., 2024), which consist of a two-stage process: (1) Adapting a model's tokenizer with tokens from the target language (Cui et al., 2023) to improve efficiency, and (2) updating model parameters through continued pre-training (CPT) to improve end task performance.

For this straightforward adaptation recipe (illustarted in Figure 1), many design choices can affect the final performance, providing trade-offs between efficiency and downstream performance. We focus on three major design choices – the choice of base LLM, the size of augmented vocabulary, and the amount of continued pretraining data. We empirically evaluate how these design choices impact the final task performances on four diverse languages (Hindi, Turkish, Arabic and Tamil) on multilingual benchmarks containing up to 7 tasks. Our goal is to provide guidance on how to build language-specific LMs based on our experimental results spanning seven base LMs.[1]

We summarize our main findings here:

- The base LM performance prior to adaptation is not always indicative of the final performance. Despite its limited initial performance, monolingual models (such as LLaMA-2) can be adapted effectively i.e. achieve comparable performance

* Authors contributed equally

[1] Code is available at: https://github.com/atutej/token-language-adaptation

Findings of the Association for Computational Linguistics: EMNLP 2024, pages 10485–10500

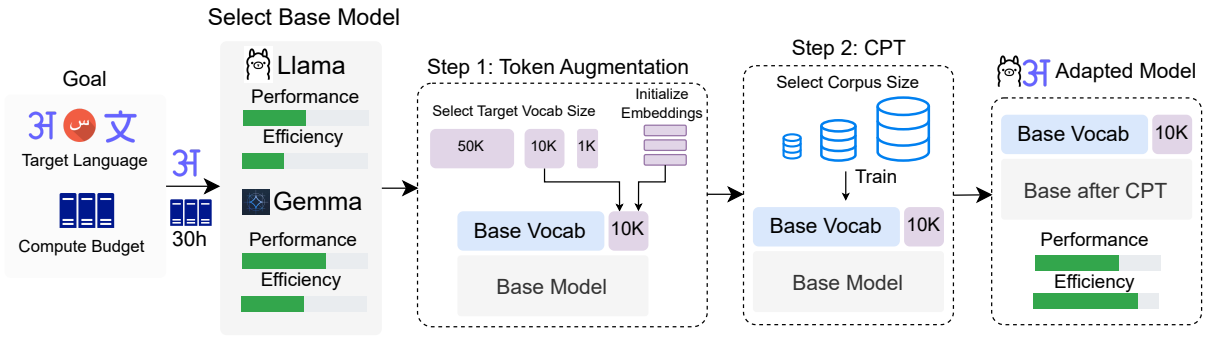


Figure 1: Building a language-specific language model. After selecting a base language model (LM), we adapt it with two main steps: 1) Token Augmentation, which primarily involves extending the tokenizer to a target vocabulary size, and 2) Continued Pre-Training on the target corpus.

to base multilingual models after training on 200M target language tokens.

- A moderate amount of vocabulary addition (10K) is sufficient to close the gap between English and low-resource languages in terms of efficiency (how many tokens are required to encode the same amount of information).
- Vocabulary extension initially drops the end task performance, but most base LMs (except the best multilingual model, Gemma-7B) can recover and improve end task performances after continued training on 200M target language tokens.
- The initialization of new token parameters is important for efficient adaptation. Initializing new token embeddings with a mean of constituent token embeddings (Liu et al., 2023), is as good as more sophisticated initialization strategies (Dobler and de Melo, 2023).
- Despite some general patterns, adaptation performance is language and base LM specific, especially with respect to the pre-training corpus of base LMs.

Together, we lay foundation for studying language-specific adaptation of existing LLMs, enabling LLM access to a wider population.

2 Related Work

Token Adaptation Early work on adapting tokenizers focused on downstream tasks (Sachidananda et al., 2021) or domains (Hiraoka et al., 2020, 2019) within a single language. Liu et al. (2023) proposed to add new tokens trained from a specific domain (mental health QA) to enhance its generation speed. Similar to our work, Dagan et al. (2024) provides a comprehensive study for optimizing tokenizers for LLMs built on code corpus. Our focus is on multilingual adaptation instead of a specific domain.

Cross-Lingual Transfer Recent efforts explore cross-lingual transfer of pre-trained English LLMs to new languages. Husain et al. (2024) transliterate Indic languages to the Latin script to transfer linguistic capabilities. Recent work (Zhao et al., 2024) showed that training on millions of target language tokens without vocabulary extension can match the end task performance of state-of-the art model trained on billions of tokens. However, this comes at the cost of inference efficiency. Most similar to ours, a line of work (Csaki et al., 2023; Cui et al., 2023; Tikhomirov and Chernyshev, 2023; Lin et al., 2024) perform vocabulary extension, showing it can improve generation efficiency on the target language. However, to the best of our knowledge, there is no unified perspective on the design choices such as the base model, vocabulary size when working with limited compute and data resources, and our study is the first that explores these aspects on multiple languages.

Previous approaches have studied ways to optimally initialize embeddings for cross-lingual transfer. Some methods have only been applied to encoder-only models (Ebrahimi and Kann, 2021; Minixhofer et al., 2022), others typically use external lexicons (Zeng et al., 2023b; Wang et al., 2022), focus primarily on Latin scripts (de Vries and Nissim, 2021), or require training a secondary model/embeddings separately (Dobler and de Melo, 2023; Ostendorff and Rehm, 2023). We compare few initialization approaches, showing that the simple mean embedding initialization works well.

Model	Vocab Size	Training Data	# Training Tokens	# Languages	# Parameters
XGLM-7.5B	256K	CC100-XL	500B	30	7.5B
Gemma-2B	256K	unknown	2T	unknown	2.5B
Gemma-7B	256K	unknown	6T	unknown	8.5B
Bloom-7.1B	251K	ROOTS	350B	46	7.1B
Phi-2	50K	Synthetic + Web	1.4T	unknown	2.7B
Mistral-7B	32K	unkown	unkown	unknown	7.2B
LLaMA-2-7B	32K	unknown	2T	unknown	6.7B
Adapted (Ours)	Base + 1K-50K	Base + subset of mC4	Base + 100M-500M	Base + 1	Base + 4M-200M

Table 1: The overview of LLMs compared in this work. For our adapted models, we add between 1K to 50K additional tokens into the vocabulary and continue training on ∼100M-500M tokens.

3 Method: Adapting LLM to a Target Language

We introduce a straightforward adaptation process – we will first generate language-specific tokens that will be added to the base vocabulary (§3.1) of the model. Then, we continue training LMs with language modeling objective on the target language corpus such that they can make use of new tokens efficiently (§3.2).

3.1 Augmenting Token Vocabulary

Generating Target Language Tokens We train a BPE sentencepiece tokenizer (Kudo and Richardson, 2018)[2] using 300K examples (i.e. documents) from the mC4 corpus (Raffel et al., 2023) on the target language, which yields a language specific vocabulary V' with a target vocab size $|V'|$. We vary the target vocab sizes from 1K to 50K.

Merging with Original Vocabulary Let $\Delta V = V' - V$ denote the non-overlapping tokens from the new vocabulary with respect to the original vocabulary V. We append these tokens to the original as $V_{\text{new}} \leftarrow V \oplus \Delta V$. Here, $\oplus$ denotes the concatenation operation, which implies that all new tokens are assigned lower priorities than those in the default vocabulary i.e. we assume that the frequency of the first 'new' token is lower than the last 'old' token in the BPE merging procedure (Sennrich et al., 2016). We preserve all the tokens from the original vocabulary, as opposed to discarding those with low-priority scores (Csaki et al., 2023). We experimented with assigning higher priorities to the extra tokens, but found that it does not lead to significant gains. The resulting effective vocabulary size can be found in Table 7 in the appendix.

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

3.2 Integrating New Tokens to the LLM

Embedding Initialization We initialize the token embeddings from the generated vocabulary ΔV as the mean embedding of its constituent tokens from the original tokenizer V, following prior work (Liu et al., 2023; Gee et al., 2022). Formally, the token embedding $E(v), \forall v \in \Delta V$ is obtained as,

$$t = \text{Tokenize}(v\,; V);\; E(v) = \frac{1}{|t|}\sum_{i}^{|t|} E(t_i) \quad (1)$$

where E is the existing token embedding. Note, for models like LLaMA-2 which use separate un-embedding (LM head) parameters, we perform the same operation separately for the un-embedding layer. We also experiment warm starting the newly introduced parameters on a tiny fraction of the dataset ($10M$ tokens) through continued pre-training (Downey et al., 2023), keeping the transformer parameters frozen and only learning the embedding and un-embedding parameters with a high learning rate (10^{-3}). We notice that this usually aids in reaching the same loss faster and can lead to small gains in the final performance.

Continued Pre-Training We perform continued pre-training on each target language with 200K examples (∼200M tokens) and 500K examples (∼500M tokens) from the mC4 corpus (Raffel et al., 2023) for larger (>6B parameters) and smaller models, respectively. The effective vocabulary sizes, and data statistics (in bytes) for the tokenizer training and continued pre-training are summarized in Table 7 in the appendix. For main experiments, we add 50K tokens when training on a larger CPT corpus (200K-500K documents). For other analyses, we use 10K tokens and train on 100K documents due to computational constraints.

Implementation Details We train on 4 A40 GPUs for a single epoch with a cosine-warmup scheduler, max sequence length 1024, with batch sizes that maximize memory usage, as presented in Table 8 in the Appendix. For 200K examples, this translates to roughly 18h of training on all 4 GPUs. We use full-finetuning in all of our experiments, since training with LoRA (Hu et al., 2022) yielded much worse performance, and only led to $1.5\times$ less compute.

4 Evaluation Setting

4.1 Experimental Goals

Language Models Table 1 summarizes the open-access base LLMs that we consider. XGLM-7.5B (Lin et al., 2021), Bloom-7.1B (Scao et al., 2023), and Gemma models (Mesnard et al., 2024) are equipped with a large vocabulary (251K-256K) that encompasses multiple languages. XGLM was trained on CC100-XL (Lin et al., 2021) on 500B tokens across 30 languages, Bloom was trained on ROOTS (Laurençon et al., 2023) on 350B tokens across ∼46 languages, while Gemma-7B and LLaMA-2-7B were trained on 6T and 2T tokens, respectively. The vocabulary of Mistral-7B and LLaMA-2 are primarily monolingual, with about 32K tokens. The smaller models, Gemma-2B and Phi-2 (Abdin et al., 2024a), were trained on 1.6T and 2T tokens, respectively. Gemma-7B variant far outperforms any existing open-sourced 7B LLMs in our evaluations on multilingual benchmarks.

Target Languages We evaluate on four languages – Hindi, Arabic, Turkish, and Tamil, which covers languages in Latin (Turkish) and non-Latin scripts (Hindi, Arabic, Tamil). Based on the performance of the base language models on downstream benchmarks, we group the languages into mid (Hindi, Arabic, Turkish) and low-resource (Tamil).

4.2 Evaluation Tasks

We select a range of multilingual benchmark datasets to cover diverse tasks. Except for machine translation and headline generation, all other benchmarks are formatted as multiple-choice tasks to facilitate evaluation. Table 9 in the Appendix reports dataset statistics.

Generation Tasks

- **Machine Translation** We evaluate on the FLORES-200 benchmark (Costa-jussà et al., 2022), which contains 1012 parallel examples in the test set. We perform 5-shot prompting with examples from the development set.
- **Headline Generation** We evaluate on XL-Sum (Hasan et al., 2021), which contains summaries and titles of news articles. We re-frame the task as generating titles from summaries, as the news articles are long. We prompt LLM with a single example from the training split.

Natural Language Understanding Tasks

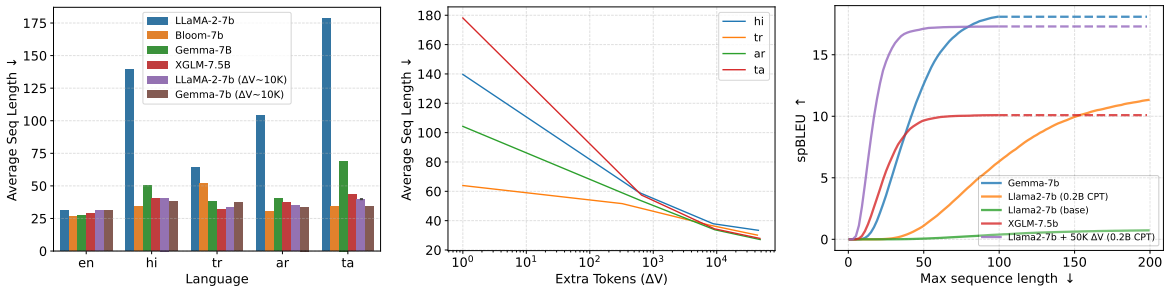
- **Knowledge Probing** We evaluate on mLAMA (Kassner et al., 2021), which contains knowledge triplets in the format ⟨object, relation, subject⟩. Following Lin et al. (2021), we convert the triplets to a template that queries for the object, such as "Matthew Perry was born in [MASK]." Apart from the ground truth, we replace [MASK] with three other candidates that belong to the same relation.
- **Natural Language Inference** We evaluate on XNLI (Conneau et al., 2018) in a 5-shot setting with the evaluation templates from prior work.[3]
- **Sentiment Analysis** We use a recently released dataset (Doddapaneni et al., 2023) of product reviews manually translated into Indic languages. The dataset contains parallel sentences in English and Indic languages. We generated an evaluation set for Turkish and Arabic through automated translation from English.[4] We evaluate in a 5-shot setting.
- **Causal/Commonsense Reasoning** We use XCOPA (Ponti et al., 2020) and XStoryCloze (Lin et al., 2021), using 5-shot in-context prompts for both tasks.

4.3 Model Inference Setting

We use the same prompt template for all models. For all multiple-choice datasets with training and evaluation splits (all datasets except for mLAMA), we select five input-output pairs from the respective training and evaluation splits to form in-context examples. The examples are selected randomly for each query to the LM, but with a fixed seed to maintain consistency across evaluations. For mLAMA, we use zero-shot prompts. The exact prompts, which include selected in-context exemplars, can be found in Appendix B.1.

[3]We use lm-evaluation-harness: https://github.com/EleutherAI/lm-evaluation-harness

[4]We use https://github.com/ssut/py-googletrans



(a) Fertility per Language & LLM pair (b) Fertility vs. Vocab Size (c) Performance vs. Fertility (Tamil)

Figure 2: Efficiency evaluation: the impact of vocabulary extension on the average sequence length. The shorter sequence length is more desirable. (a) extending the vocabulary with 10K tokens makes the token length substantially shorter, on par with that of English. (b) Adding more tokens continue to improve the sequence length with diminishing returns. (c) Performance (spBLEU) on FLORES machine translation versus maximum generation cutoff length; adapted models require 2× less tokens to achieve high performance.

For multiple-choice tasks, we select the continuation with the highest byte-length normalized log-probabilities, which is tokenizer agnostic (Gao et al., 2023). For generation tasks, we sample greedily, with a maximum of 200 tokens for translation and 50 tokens for summarization.

4.4 Evaluation Metrics

Efficiency Metrics As we do not make changes to LLM architecture except for its tokenizer, our efficiency metric will focus on the number of tokens required to convey the same amount of information per language, following prior work (Ahia et al., 2023). Specifically, we define the fertility as the average number of tokens required to encode a given text. This diverges from earlier work (Rust et al., 2021) which defines fertility as the average number of sub-words per given word and make evaluation independent of word segmentation.

For generation tasks, we additionally measure (%Gen), the percentage of generated tokens that belong to the newly added vocabulary, ΔV, and the number of examples processed per second (throughput). We define throughput as the number of examples processed by the model per second.

Task Performance Metrics For the text-understanding benchmarks (multiple-choice questions), we measure the accuracy. For machine translation and summarization, we report spBLEU (Goyal et al., 2022), a universal version of BLEU that is comparable across languages.

5 Results

We report performances on two axis – efficiency (fertility) and end task performance.

5.1 Efficiency Analysis

Vocabulary augmentation effectively improves fertility in low-resource languages We compare the fertility between tokenizers by measuring the average sequence length on target sentences from the FLORES benchmark. Figure 2a shows a significant disparity in fertility between English and other languages, especially low-resource, before vocabulary adaptation. The disparity is more pronounced on primarily monolingual LMs. Similar disparity has been observed in commercial language models with multilingual capabilities (Ahia et al., 2023). Augmenting the base vocabulary of these models with ∼10K language-specific tokens significantly mitigates the disparity, providing fertility of the target language that is on-par with that of English. For low-resource language (e.g., Tamil), even multilingual model (Gemma-7B) shows significant gain in fertility after vocabulary extension.

Relationship between the vocab size and fertility Figure 2b reports the relationship between the number of added tokens $|\Delta V|$ and fertility when extending LLaMA-2's vocabulary. We observe that fertility on the target-language improves with increasing vocabulary size, but with diminishing gains as the extra token increases. Adding ∼1K language-specific tokens doubles the fertility on average. The gains are more pronounced for non-latin scripts (hi, ar, ta), with up to 3x increase after adding ∼1K tokens.

5.2 End Task Performance

We have found that fertility can be improved with relatively modest amount of vocabularly extension. But can LLM make use of newly added vocabulary

Lang	Base model	Task						
		FLORES	**XLSUM**	**MLAMA**	**Sentiment**	**XStoryCloze**	**XNLI**	**XCOPA**
	Random Guess	-	-	25.00	50.00	50.00	33.00	50.00
hi/tr	Gemma-7B	**32.26**/**26.64**	**14.60**/15.21	**50.05**/56.84	**95.20**/**97.20**	**70.62**/-	**41.65**/42.25	-/**69.00**
	Bloom-7.1B	21.32/1.23	7.91/7.87	47.21/40.52	94.00/64.70	64.99/-	**41.65**/34.02	-/51.60
	XGLM-7.5B	18.12/16.75	8.66/6.50	43.21/55.96	82.40/66.90	61.22/-	40.72/41.49	-/60.00
	LLaMA-2-7B	8.17/6.61	10.74/12.78	38.92/48.42	91.60/90.10	57.18/-	36.63/41.37	-/53.40
	LLaMA-2 (ΔV=50K, CPT)	28.15/20.95	13.70/**16.03**	44.45/**61.72**	87.60/85.10	67.90/-	39.36/40.48	-/61.80
ta/ar	Gemma-7B	**18.08**/**26.99**	14.51/17.59	**44.65**/52.67	**96.70**/**97.60**	-/**68.83**	-/**39.60**	56.20/-
	Bloom-7.1B	8.74/20.40	10.91/13.51	41.41/**53.55**	64.70/85.40	-/62.67	-/37.51	54.80/-
	XGLM-7.5B	10.17/12.23	3.86/6.84	37.19/47.18	76.10/71.10	-/58.37	-/36.18	51.60/-
	LLaMA-2-7B	5.89/0.74	12.14/3.95	40.64/36.81	72.20/62.10	53.28/-	36.55/-	-/48.60
	LLaMA-2 (ΔV=50K, CPT)	17.29/22.33	**16.37**/**17.66**	41.55/47.00	93.20/89.70	-/64.59	-/35.82	**58.40**/-

Table 2: Comparing adapted monolinugal model (LLaMA-2-7B) against state-of-the-art multilingual models. We report spBLEU for FLORES (en → [lang]) dataset and accuracy for all other datasets. **Bold** values indicate the best performance in for each dataset while underlined values indicate the second best number. In this table, $\Delta V = \sim 50K$ (i.e. 50K added tokens) and CPT done for 200M tokens. Our adapted models will have a separate checkpoint per language, while others use the same checkpoint for all languages.

	en → hi/en → tr			en → ar/en → ta		
Model	**Throughput ↑**	**Fertility↓**	**%Gen**	**Throughput ↑**	**Fertility↓**	**%Gen**
XGLM	0.71/0.82	39.85/**34.81**	-	0.71/0.57	39.98/48.84	-
Gemma-7B	0.47/0.63	52.34/40.04	-	0.63/0.32	38.06/77.14	-
LLaMA-2 + CPT	0.18/0.31	154.52/93.02	-	0.24/0.14	116.71/192.13	-
LLaMA-2 (ΔV=50K, CPT)	**0.85**/**0.84**	**37.49**/39.04	81.72/62.96	**1.06**/**0.89**	**27.59**/**32.87**	79.85/70.43

Table 3: Efficiency comparisons on FLORES machine translation task. % Gen indicates the % of tokens generated that belong to the extra vocabulary ΔV. Fertility is the average sequence length of generations. Adapted LLaMA-2 achieves throughput that is up to 68% higher than multilingual models.

tokens efficiently, given they were unseen during its pre-training? To answer this question, we discuss end task performances.

Adapted English models can match the performance of base multilingual models Multilingual LLMs often show stronger performance on a wide range of language compared to primarily English LLMs. However, state-of-the-art English models are released more frequently with various configurations (model sizes, architecture, focus domains) than their multilingual equivalents. We study whether adapting these predominantly English models through vocabulary extension and continued pre-training (CPT) can enable them to match the performance of multilingual models. In Table 2, we compare the adapted LLaMA-2-7B model against XGLM-7.5B, Gemma-7B, and Bloom-7.1B. Notably, the recently released Gemma-7B model demonstrates superior performance in all tasks across four languages.

LLaMA model performance improves substantially through pre-training on a relatively small target-language corpus. Our adapted LLaMA model, LLaMA-2-7B (ΔV=50K, CPT), exhibits highly competitive performance with respect to Gemma 7B, particularly on the low-resource language (Tamil), with 1.5 billion fewer parameters. As shown in Figure 2c, LLaMA-2-7B (ΔV=50K, CPT) competes with Gemma-7B using 2× less tokens. Additionally, as illustrated in Table 3, the adapted model achieves up to 8 times higher throughput compared to LLaMA without vocabulary extension and also substantially higher than Gemma 7B. These enhancements are more pronounced for non-Latin and low-resource languages (Hindi, Arabic, Tamil) compared to Latin

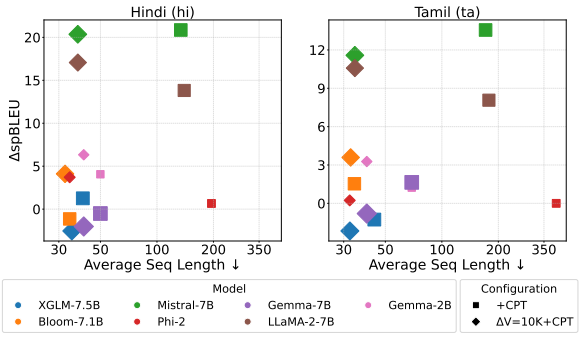


Figure 3: Change in performance (ΔspBLEU) after adaptation. We report average performance across all benchmarks for two language pairs hi⁄ta, with continued training on 100K examples and vocabulary extension of ΔV=10K tokens. Larger models are represented with bigger markers. Absolute numbers on all benchmarks are provided in Table 12 and Table 13 in Appendix C.

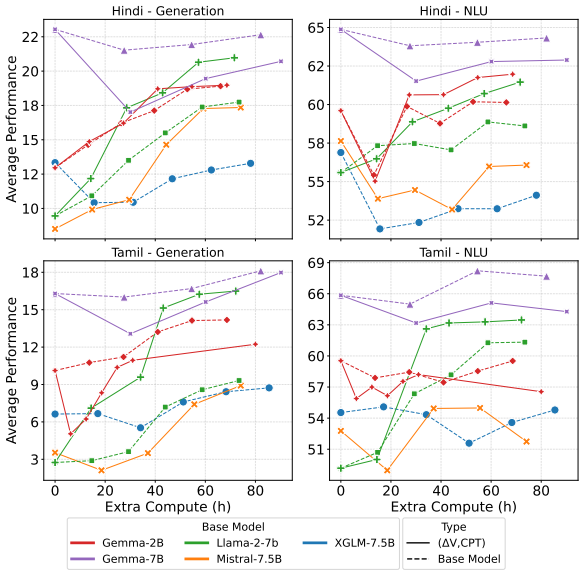


Figure 4: Adapted LLM's performance on Hindi/Tamil on generation and understanding benchmarks with increasing compute (measured in terms of hours per GPU). For models with extended vocabulary, ΔV=50K.

languages (Turkish). Moreover, adapted monolingual models produce up to 1.5 times× smaller sequences than XGLM and Gemma models.

6 Analysis

Can multilingual models benefit from the same adaptation recipe? Figure 3 illustrates the change in performance (ΔspBLEU) on the FLORES generation task observed when adapting various monolingual and multilingual models for Hindi (hi) and Tamil (ta). Here, we fix the adaptation recipe: we train on 100K examples, both with and without vocabulary augmentation (∼10K tokens). We compare four kinds of base models - large monolingual (Mistral-7B, LLaMA-2-7B), small monolingual (Phi-2), large multilingual (XGLM-7.5B, Bloom-7.1B, Gemma-7B), and small multilingual (Gemma-2B).

The lack of end task performance gains: Phi-2 and Gemma-7B Larger monolingual models (LLaMA and Mistral) improve up to 12× over their base variants. Here, the more capable English model (Mistral) performs better after adaptation for both mid and low resource settings. The smaller, yet highly competitive, monolingual model i.e. Phi-2 (Abdin et al., 2024a,b) exhibits minimal improvement, particularly on Tamil. Phi-2 is the only model in our comparisons that was trained on a curated, high quality monolingual data. This suggests that adapting monolingual model trained on a highly curated data might be more challenging.

Smaller multilingual models like Gemma-2B still exhibit up to 56% relative improvement, and show competitive end-task performance after adaptation. Gemma-7B, which was trained on extensive amounts of data (6T tokens), does not show notable performance enhancements from either continued pretraining (CPT) or vocabulary extension. This may indicate that its extensive pretraining saturates multilingual capabilities, rendering additional improvements through CPT or vocabulary expansion redundant. Similarly, XGLM shows limited benefit from these adaptations, likely due to its pre-training dataset that mirrors our CPT corpus – suggesting that when the training data is already aligned with the adaptation data, further modifications may not yield substantial performance gains. In contrast, BLOOM, which is trained on a curated corpus of 46 languages (Laurençon et al., 2023), demonstrates relative improvements of up to 40%.

Training a smaller model with more data vs. a larger model with less data. As shown in Figure 4, this choice depends heavily on the target language, compute budget, and the type of base model. For shorter training periods (<30h), the smaller multilingual model (Gemma-2B) is highly effective for both mid-resource (Hindi) and low-resource (Tamil) languages, making it a viable choice in scenarios where computational resources are limiting factors. While Gemma-7B achieves the highest performance across all tasks and languages, it consumes more computational overhead

CPT data	V'	#params	FLORES	XLSUM	MLAMA	Sentiment	XStoryCloze	XNLI
100K	1K	6.7B	23.46	8.65	43.44	90.20	**67.24**	40.00
	10K	6.8B	**25.24**	11.99	**44.14**	**92.10**	65.65	41.04
	50K	7.1B	25.02	**12.36**	42.45	91.50	67.11	**41.61**
200K	1K	6.7B	26.50	9.23	45.08	91.20	65.65	39.40
	10K	6.8B	27.31	**12.45**	**45.62**	86.60	65.12	40.48
	50K	7.1B	**27.86**	12.36	43.87	**94.10**	**67.84**	**41.41**

Table 4: Performance on Hindi (LLama-2-7B) with increasing vocabulary size and CPT data (# examples). With larger amount of data (200K), larger vocab size (50K) leads to performance gain, while with smaller amount of data (100K), smaller vocab (10K) often leads better performance.

Model	FLORES	XLSUM	MLAMA	Sentiment	XStoryCloze	XNLI
`LLaMA-2-7B (base)`	8.17	10.74	38.92	91.60	57.18	36.63
`Random-Init`	17.82	10.59	40.37	91.90	63.34	39.88
`Random-Tok-Emb`	17.01	9.22	39.09	78.90	62.67	39.68
`Mean`	**25.24**	11.99	44.14	92.10	65.65	**41.04**
`FOCUS`	24.33	**12.79**	43.40	**93.20**	66.78	38.59
`Learned-Emb`	24.59	12.41	**44.67**	86.50	**67.50**	39.92

Table 5: Results on Hindi benchmarks (LLaMA-2-7B) with varying embedding initialization strategies (ΔV=10K). Simple mean of constituent tokens performs competitively with respect to more complex methods.

due to its larger size (up to 1B more than any other model we consider), making it less practical for resource-limited settings.

With longer training times (>40h), LLaMA-2-7B surpasses Gemma-2B on generation tasks for Hindi (Figure 4), and across all tasks for Tamil. A larger target vocabulary size of 50K seems to hinder performance in the stronger English model (Mistral) as opposed to a smaller vocabulary size of 10K (Figure 3). LLaMA-2-7B offers a good balance between adaptation computational efficiency and performance. This makes LLaMA-2-7B suitable for low-resource languages, particularly on generation tasks, when longer training periods are possible, but with a trade-off of increased inference time.

Size of augmented vocabulary can be scaled proportionally to CPT data We compare performance of our model variants with increasing target vocabulary size in Table 4. In terms of efficiency, the bigger the vocabulary size, the fewer the number of tokens that are needed to encode the same amount of information. But how would it impact end task performance? We observe a more mixed result here: when the models are trained on smaller amounts of data (100K examples), there is no significant gain from adding more than 10K tokens to the base vocabulary. However, on doubling the training data, performance continues to grow with vocabulary size. Our findings align with Dagan et al. (2024), which studied vocabulary extension to adapt LLM to a code domain, that larger vocabulary sizes do not decrease downstream task performance when fine-tuning on billions of tokens. In our experiments, we observe that additional training only in the order of million tokens is sufficient for adaptation of monolingual models.

Mean embedding initialization is simple and effective We present the results of five different initialization strategies in Table 5. `Random-Init` initializes the new token embeddings with random values, while `Random-Tok-Emb` uses the embedding of a randomly selected existing token for initialization. In the `Learned-Emb` strategy, we freeze all the model layers except for the embedding layer and train on 10% of the CPT data before switching to full fine-tuning. `FOCUS` (Dobler and de Melo, 2023) computes embeddings based on semantic similarity within an auxiliary static token embedding space. We observe that `FOCUS`, `Mean`, and `Learned-Emb` outperform the random initialization techniques. Notably, the simplicity of `Mean`, combined with its competitive efficacy, renders it an appealing choice for initialization. `Learned-Emb` suffers a performance drop on sentiment analysis, where the performance was high on this benchmark to begin with, and `Learned-Emb` might initialize embeddings that are farther away in the original embedding space as opposed to `Mean` and `FOCUS`. Our

Model	MLAMA	Sent	XStory	XNLI
LLaMA-2-7B	73.19	98.60	89.08	52.73
LLaMA-2-7B (ΔV=10K, CPT)	62.18	92.10	82.46	52.97
ΔPerf.	-11.01	-6.50	3.02	0.24

Table 6: Evaluation of catastrophic forgetting on English benchmarks after continued pre-training on Hindi (hi) with 100K examples.

findings reinforce prior work showing that vocabulary initialization has big impact on downstream performances (Minixhofer et al., 2022; Dobler and de Melo, 2023).

Does LLM lose performance in English after adaptation? Partially. Table 6 contrasts the downstream performance of our adapted model (ΔV=10, CPT) with that of the base model (LLaMA-2-7B) on the source language (English). We observe that performance on these tasks drops by 3.56%, with the highest drop of 11% on the knowledge probing tasks (MLAMA). Interestingly, we note that the cross-lingual task on the target language (English (en) $\rightarrow$ Hindi (hi)) is not significantly impacted by catastrophic forgetting, and in fact, improves after adaptation (Table 2).

7 Conclusion

In this paper, we presented a systematic study on the design choices involved in adapting LLMs to specific target languages. We explored the strategic choices in the adaptation process, such as the necessity of vocabulary extension, the addition of a large pool of language-specific tokens, and the criticality of initializing new parameters effectively. We contextualize these choices with respect to the end goal, such as the target language and a specified compute budget. Through comprehensive experiments on up to four languages and seven base language models, we showed that the efficiency of the model on a target language can be improved by a simple vocabulary augmentation step followed by further training. We further show that under the same adaptation strategy, adapted smaller multilingual models can be as good as their larger counterparts, and larger monolingual LMs can perform almost as good as multilingual LMs with relatively minimal amounts of target language data.

Limitations

In this study, we are restricted to analyzing four languages due to computational limitations. In future research, we aim to explore a much wider range of languages, and a larger set of base models. We also did not explore trends on families of monolingual models of varying sizes. The vocabulary extension approach we study is limited to a straightforward union of vocabularies, and we did not investigate the factors in the BPE training corpus that might affect the generated tokens. We also did not investigate alternative tokenization approaches besides BPE.

Acknowledgments

We would like to thank Adam Klivans for providing compute resources. We also thank the members of UT NLP community and reviewers for feedback. This work was in part supported by Cisco Research. Any opinions, findings and conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of Cisco Research.

References

Marah Abdin, Jyoti Aneja, Sebastien Bubeck, Caio César Teodoro Mendes, Weizhu Chen, Allie Del Giorno, Ronen Eldan, Sivakanth Gopi, Suriya Gunasekar, Mojan Javaheripi, et al. 2024a. Phi-2: The surprising power of small language models — microsoft.com.

Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Qin Cai, Martin Cai, Caio César Teodoro Mendes, Weizhu Chen, et al. 2024b. Phi-3 technical report: A highly capable language model locally on your phone. *Preprint*, arXiv:2404.14219.

Orevaoghene Ahia, Sachin Kumar, Hila Gonen, Jungo Kasai, David R Mortensen, Noah A Smith, and Yulia Tsvetkov. 2023. Do all languages cost the same? tokenization in the era of commercial language models. *arXiv preprint arXiv:2305.13707*.

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.

Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume

Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No language left behind: Scaling human-centered machine translation. *Preprint*, arXiv:2207.04672.

Zoltan Csaki, Pian Pawakapan, Urmish Thakker, and Qiantong Xu. 2023. Efficiently adapting pretrained language models to new languages. *arXiv preprint arXiv:2311.05741*.

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

Gautier Dagan, Gabriel Synnaeve, and Baptiste Rozière. 2024. Getting the most out of your tokenizer for pre-training and domain adaptation. *Preprint*, arXiv:2402.01035.

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.

Sumanth Doddapaneni, Rahul Aralikatte, Gowtham Ramesh, Shreya Goyal, Mitesh M. Khapra, Anoop Kunchukuttan, and Pratyush Kumar. 2023. Towards leaving no Indic language behind: Building monolingual corpora, benchmark and models for Indic languages. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12402–12426, Toronto, Canada. Association for Computational Linguistics.

CM Downey, Terra Blevins, Nora Goldfine, and Shane Steinert-Threlkeld. 2023. Embedding structure matters: Comparing methods to adapt multilingual vocabularies to new languages. *arXiv preprint arXiv:2309.04679*.

Abteen Ebrahimi and Katharina Kann. 2021. How to adapt your pretrained multilingual model to 1600 languages. 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 4555–4567, Online. Association for Computational Linguistics.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2023. A framework for few-shot language model evaluation.

Leonidas Gee, Andrea Zugarini, Leonardo Rigutini, and Paolo Torroni. 2022. Fast vocabulary transfer for language model compression. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 409–416, Abu Dhabi, UAE. Association for Computational Linguistics.

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

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.

Tatsuya Hiraoka, Hiroyuki Shindo, and Yuji Matsumoto. 2019. Stochastic tokenization with a language model for neural text classification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1620–1629, Florence, Italy. Association for Computational Linguistics.

Tatsuya Hiraoka, Sho Takase, Kei Uchiumi, Atsushi Keyaki, and Naoaki Okazaki. 2020. Optimizing word segmentation for downstream task. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1341–1351.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Jaavid Aktar Husain, Raj Dabre, Aswanth Kumar, Ratish Puduppully, and Anoop Kunchukuttan. 2024. Romansetu: Efficiently unlocking multilingual capabilities of large language models models via romanization. *Preprint*, arXiv:2401.14280.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao,

Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Nora Kassner, Philipp Dufter, and Hinrich Schütze. 2021. Multilingual LAMA: Investigating knowledge in multilingual pretrained language models. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3250–3258, Online. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226*.

Hugo Laurençon, Lucile Saulnier, Thomas Wang, Christopher Akiki, Albert Villanova del Moral, Teven Le Scao, Leandro Von Werra, Chenghao Mou, Eduardo González Ponferrada, Huu Nguyen, Jörg Frohberg, Mario Šaško, Quentin Lhoest, Angelina McMillan-Major, Gerard Dupont, Stella Biderman, Anna Rogers, Loubna Ben allal, Francesco De Toni, Giada Pistilli, Olivier Nguyen, Somaieh Nikpoor, Maraim Masoud, Pierre Colombo, Javier de la Rosa, Paulo Villegas, Tristan Thrush, Shayne Longpre, Sebastian Nagel, Leon Weber, Manuel Muñoz, Jian Zhu, Daniel Van Strien, Zaid Alyafeai, Khalid Almubarak, Minh Chien Vu, Itziar Gonzalez-Dios, Aitor Soroa, Kyle Lo, Manan Dey, Pedro Ortiz Suarez, Aaron Gokaslan, Shamik Bose, David Adelani, Long Phan, Hieu Tran, Ian Yu, Suhas Pai, Jenny Chim, Violette Lepercq, Suzana Ilic, Margaret Mitchell, Sasha Alexandra Luccioni, and Yacine Jernite. 2023. The bigscience roots corpus: A 1.6tb composite multilingual dataset. *Preprint*, arXiv:2303.03915.

Aaron Levine, Connie Huang, Chenguang Wang, Eduardo Batista, Ewa Szymanska, Hongyi Ding, Hou Wei Chou, Jean-François Pessiot, Johanes Effendi, Justin Chiu, Kai Torben Ohlhus, Karan Chopra, Keiji Shinzato, Koji Murakami, Lee Xiong, Lei Chen, Maki Kubota, Maksim Tkachenko, Miroku Lee, Naoki Takahashi, Prathyusha Jwalapuram, Ryutaro Tatsushima, Saurabh Jain, Sunil Kumar Yadav, Ting Cai, Wei-Te Chen, Yandi Xia, Yuki Nakayama, and Yutaka Higashiyama. 2024. Rakutenai-7b: Extending large language models for japanese. *Preprint*, arXiv:2403.15484.

Peiqin Lin, Shaoxiong Ji, Jörg Tiedemann, André F. T. Martins, and Hinrich Schütze. 2024. Mala-500: Massive language adaptation of large language models. *Preprint*, arXiv:2401.13303.

Xi Victoria Lin, Todor Mihaylov, Mikel Artetxe, Tianlu Wang, Shuohui Chen, Daniel Simig, Myle Ott, Naman Goyal, Shruti Bhosale, Jingfei Du, et al. 2021. Few-shot learning with multilingual language models. *arXiv preprint arXiv:2112.10668*.

Siyang Liu, Naihao Deng, Sahand Sabour, Yilin Jia, Minlie Huang, and Rada Mihalcea. 2023. Enhancing long-form text generation in mental health with task-adaptive tokenization. *arXiv preprint arXiv:2310.05317*.

Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, Amélie Héliou, Andrea Tacchetti, Anna Bulanova, Antonia Paterson, Beth Tsai, Bobak Shahriari, Charline Le Lan, Christopher A. Choquette-Choo, Clément Crepy, Daniel Cer, Daphne Ippolito, David Reid, Elena Buchatskaya, Eric Ni, Eric Noland, Geng Yan, George Tucker, George-Christian Muraru, Grigory Rozhdestvenskiy, Henryk Michalewski, Ian Tenney, Ivan Grishchenko, Jacob Austin, James Keeling, Jane Labanowski, Jean-Baptiste Lespiau, Jeff Stanway, Jenny Brennan, Jeremy Chen, Johan Ferret, Justin Chiu, Justin Mao-Jones, Katherine Lee, Kathy Yu, Katie Millican, Lars Lowe Sjoesund, Lisa Lee, Lucas Dixon, Machel Reid, Maciej Mikuła, Mateo Wirth, Michael Sharman, Nikolai Chinaev, Nithum Thain, Olivier Bachem, Oscar Chang, Oscar Wahltinez, Paige Bailey, Paul Michel, Petko Yotov, Pier Giuseppe Sessa, Rahma Chaabouni, Ramona Comanescu, Reena Jana, Rohan Anil, Ross McIlroy, Ruibo Liu, Ryan Mullins, Samuel L Smith, Sebastian Borgeaud, Sertan Girgin, Sholto Douglas, Shree Pandya, Siamak Shakeri, Soham De, Ted Klimenko, Tom Hennigan, Vlad Feinberg, Wojciech Stokowiec, Yu hui Chen, Zafarali Ahmed, Zhitao Gong, Tris Warkentin, Ludovic Peran, Minh Giang, Clément Farabet, Oriol Vinyals, Jeff Dean, Koray Kavukcuoglu, Demis Hassabis, Zoubin Ghahramani, Douglas Eck, Joelle Barral, Fernando Pereira, Eli Collins, Armand Joulin, Noah Fiedel, Evan Senter, Alek Andreev, and Kathleen Kenealy. 2024. Gemma: Open models based on gemini research and technology. *Preprint*, arXiv:2403.08295.

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.

Martin Müller and Florian Laurent. 2022. Cedille: A large autoregressive french language model. *Preprint*, arXiv:2202.03371.

Malte Ostendorff and Georg Rehm. 2023. Efficient language model training through cross-lingual and progressive transfer learning. *Preprint*, arXiv:2301.09626.

Edoardo Maria Ponti, Goran Glavaš, Olga Majewska, Qianchu Liu, Ivan Vulić, and Anna Korhonen. 2020. XCOPA: A multilingual dataset for causal commonsense reasoning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language*

Processing (EMNLP), pages 2362–2376, Online. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2023. Exploring the limits of transfer learning with a unified text-to-text transformer. *Preprint*, arXiv:1910.10683.

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.

Vin Sachidananda, Jason Kessler, and Yi-An Lai. 2021. Efficient domain adaptation of language models via adaptive tokenization. In *Proceedings of the Second Workshop on Simple and Efficient Natural Language Processing*, pages 155–165, Virtual. Association for Computational Linguistics.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, and et al. 2023. Bloom: A 176b-parameter open-access multilingual language model. *Preprint*, arXiv:2211.05100.

Neha Sengupta, Sunil Kumar Sahu, Bokang Jia, Satheesh Katipomu, Haonan Li, Fajri Koto, William Marshall, Gurpreet Gosal, Cynthia Liu, Zhiming Chen, Osama Mohammed Afzal, Samta Kamboj, Onkar Pandit, Rahul Pal, Lalit Pradhan, Zain Muhammad Mujahid, Massa Baali, Xudong Han, Sondos Mahmoud Bsharat, Alham Fikri Aji, Zhiqiang Shen, Zhengzhong Liu, Natalia Vassilieva, Joel Hestness, Andy Hock, Andrew Feldman, Jonathan Lee, Andrew Jackson, Hector Xuguang Ren, Preslav Nakov, Timothy Baldwin, and Eric Xing. 2023. Jais and jais-chat: Arabic-centric foundation and instruction-tuned open generative large language models. *Preprint*, arXiv:2308.16149.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. *Preprint*, arXiv:1508.07909.

Mikhail Tikhomirov and Daniil Chernyshev. 2023. Impact of tokenization on llama russian adaptation. *Preprint*, arXiv:2312.02598.

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

Xinyi Wang, Sebastian Ruder, and Graham Neubig. 2022. Expanding pretrained models to thousands more languages via lexicon-based adaptation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 863–877, Dublin, Ireland. Association for Computational Linguistics.

Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, Weng Lam Tam, Zixuan Ma, Yufei Xue, Jidong Zhai, Wenguang Chen, Peng Zhang, Yuxiao Dong, and Jie Tang. 2023a. Glm-130b: An open bilingual pre-trained model. *Preprint*, arXiv:2210.02414.

Qingcheng Zeng, Lucas Garay, Peilin Zhou, Dading Chong, Yining Hua, Jiageng Wu, Yikang Pan, Han Zhou, Rob Voigt, and Jie Yang. 2023b. Greenplm: Cross-lingual transfer of monolingual pretrained language models at almost no cost. *Preprint*, arXiv:2211.06993.

Jun Zhao, Zhihao Zhang, Luhui Gao, Qi Zhang, Tao Gui, and Xuanjing Huang. 2024. Llama beyond english: An empirical study on language capability transfer. *Preprint*, arXiv:2401.01055.

Ahmet Üstün, Viraat Aryabumi, Zheng-Xin Yong, Wei-Yin Ko, Daniel D'souza, Gbemileke Onilude, Neel Bhandari, Shivalika Singh, Hui-Lee Ooi, Amr Kayid, Freddie Vargus, Phil Blunsom, Shayne Longpre, Niklas Muennighoff, Marzieh Fadaee, Julia Kreutzer, and Sara Hooker. 2024. Aya model: An instruction finetuned open-access multilingual language model. *Preprint*, arXiv:2402.07827.

Appendix

The appendix is organized as follows:

- In Appendix A, we present the training setup and other details, such as CPT dataset statistics.
- In Appendix B, we describe evaluation data statistics and evaluation setup such as prompt templates.
- In Appendix C, we present additional results and experiments.

A Training Setup

Language	#Bytes: Tok	|V'|	$\|V_{new}\|$	#Bytes: CPT
Hindi (hi)	1.4B	1K	32,613	0.5B
		10K	40,816	0.8B
		50K	77,338	0.9B
Turkish (tr)	0.9B	1K	32,331	0.2B
		10K	39,516	0.3B
		50K	75,773	0.4B
Arabic (ar)	1.4B	1K	32,638	0.4B
		10K	41,337	0.6B
		50K	80,348	0.7B
Tamil (ta)	2.6B	1K	32,660	0.6B
		10K	41,233	1.1B
		50K	79,967	1.3B

Table 7: Our training dataset statistics. #Bytes: Tok indicates the total number of bytes used to train the language-specific tokenizers, and #Bytes: CPT indicate the effective number of bytes seen by the model during continued pre-training (CPT). $|V'|$ and $|V_{\text{new}}|$ indicate the vocab sizes of the language-specific vocabularies and merged vocabularies (§3.1), respectively.

Model	Learning Rate	Batch Size
LLaMA-2-7B	6E-05	8
Gemma-7B	3E-06	4
Gemma-2B	9E-05	16
Mistral-7B	3E-05	6
Bloom-7.1B	6E-05	8
XGLM-7.5B	3E-05	6
Phi-2	9E-05	16

Table 8: CPT hyperparameter setup for each model, along with per-device batch size. We manually tune hyperparameters. Due to compute limitations, we report numbers on single runs.

A.1 FOCUS Implementation

We follow the codebase in https://github.com/konstantinjdobler/focus, which uses pre-trained fastText embeddings to initialize new embeddings.

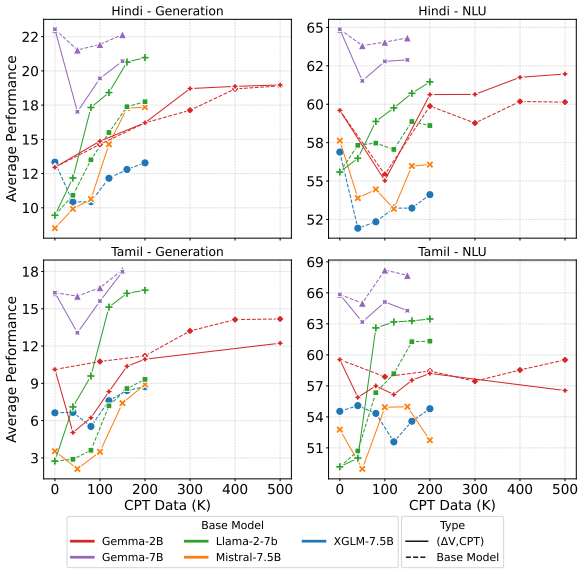


Figure 5: Performance variation on Hindi benchmarks with increasing CPT data (#examples). For models with extended vocabulary (LLaMA, Mistral), ΔV=50K.

B Evaluation Settings

The statistics for each evaluation dataset are reported in Table 9.

	License	hi	tr	ar	ta
FLORES	CC-BY-SA-4.0	1012	1012	1012	1012
MLAMA	CC-BY-NC-SA-4.0	8570	14209	19354	7223
Sentiment	CC-0	1000	1000	1000	1000
XStoryCloze	CC-BY-SA-4.0	1511	-	1511	-
XNLI	CC-BY-NC-4.0	5010	5010	5010	-
XCOPA	CC-BY-4.0	-	500	-	500
XLSUM	CC-BY-NC-SA-4.0	8847	3397	4689	2027

Table 9: Datasets: license information and statistics – number of test examples for each benchmark & language.

B.1 Prompt Templates

For each task, we formulate the prompt as follows. Examples are provided in Figure 6.

Sentiment: *"sentence: [sentiment]"*, *sentiment* ∈ {*Positive, Negative*}

XNLI: *"premise. [connector]? hypothesis"*, *connector* ∈ {*Yes, No, Also*} (in the target language)

XStoryCloze: *"sentence, [continuation]" continuation* ∈ $\mathcal{C}$, set of candidate continuations.

XCOPA: *"premise. [continuation]"*, *continuation* ∈ $\mathcal{C}$, set of candidate continuations.

FLORES: *"English: [English Text], [Target*

	Bloom-7.1B	Bloom-7.1B (CPT)	Bloom-7.1B ΔV=50K, CPT
Fertility	52.34	52.34	39.21
FLORES	1.23	8.96	12.26
MLAMA	40.52	58.93	58.25
Sentiment	64.70	53.80	77.60
XNLI	34.02	40.08	37.79
XCOPA	51.60	55.40	54.00
Average	40.74	44.92	46.52

Table 10: Performance on Bloom-7.1B after adaptation on Turkish (unseen language). We train further on 100K examples.

Method	FLORES	MLAMA	XCOPA	Sent	XLSUM
FOCUS	10.61	38.06	56.20	93.10	16.76
Mean	11.32	42.12	55.00	93.10	16.61

Table 11: Performance on Tamil Benchmarks (LLaMA-2-7B) with varying embedding initializaiton strategies (ΔV=10K.

Language]: [Translated Text]"
XLSUM: *"Summary: [Summary Text], Title: [Title Text]"*

C Additional Experiments

Impact of CPT and vocabulary extension on various multilingual and monolingual models In Table 12 and Table 13 we present the full performance of all base models after adaptation – without and with vocabulary augmentation. We observe results that are consistent with what we discuss in subsection 5.2. Adapted primarily monlingual models (LLaMA, Mistral) show significant gains and match, if not surpass, most multilingual variants.

Given a fixed amount of data, train a smaller multilingual model or larger monolingual model? As shown in Figure 5, when given a fixed amount of data rather than a compute budget, training a larger model (either monolingual i.e LLaMA-2 or Gemma-7B) triumphs over tuning Gemma-2B, which needs more data. Again, the more capable English model (Mistral) does not do very well on a large amount of added tokens (50K).

Does Vocabulary Extension with CPT benefit multilingual models on unseen languages? We report performance on Bloom-7.1B with vocabulary extension and further training in Table 10. As shown in Figure 2a, the fertility on Turkish for Bloom is sub-optimal as it is an unseen language. We observe that vocabulary augmentation followed by CPT improves ferility by 25%. On the generation tasks, we observe a relative performance improvement of 36%, and 3% improvement on all tasks on average as compared to just CPT.

Benchmark		
XCOPA (ta)	Example	அந்த பெண் மோசமான மனநிலையில் இருந்தாள எனவே அவள் தன் தோழியிடம் அவளைத் தனிமையில் விடுமாறு கூறினாள்
	Translation	The girl was in a bad mood, so she asked her friend to leave her alone
Sentiment (hi)	Example	यह आपको अपने पसंदीदा पॉडकास्ट को डाउनलोड नहीं करने देती है, इसलिए आपको ऐप का इस्तेमाल करते समय हर समय डेटा ऑन रखना होगा।: Negative
	Translation	It doesn't let you download your favorite podcasts, so you'll have to keep data on at all times while using the app: Negative
XNLI (tr)	Example	Bu önemli, doğru? Böylece, Haber hikayesi için büyük
	Translation	This is important, right? Thus, great for News story.
XStoryCloze (hi)	Example	जमि ने पालक वाली कुकीज़ बनाई. उसने एक कुकी का दाम पांच डॉलर रखा. कसी ने भी नहीं ख़रीदा. हालांकि, बलिकुल आखरि में एक आदमी ने 20 कुकी मांगी. वह बहुत हैरान हो गया लेकनि आभारी था.
	Translation	Jim made spinach cookies. He priced one cookie at five dollars. Nobody bought it. However, at the very end, a man asked for 20 cookies. He was very surprised but grateful.
XLSUM (hi)	Example	Summary: ऑस्ट्रेलयिा के ग्लेन मैकग्रॉ टेस्ट क्रकिेट में सबसे सफल तेज़ गेंदबाज़ बन गए हैं. वशि्व एकादश के ख़लिाफ़ चल रहे सुपर टेस्ट मैच के दूसरे दनि मैकग्रॉ ने वेस्टइंडीज़ के तेज़ गेंदबाज़ कर्टनी वॉल्श का रकिॉर्ड तोड़ दयिा. Title: मैकग्रॉ के करयिर में एक और नगीन.
	Translation	Summary: Australia's Glenn McGrath has become the most successful fast bowler in Test cricket. On the second day of the ongoing Super Test match against World XI, McGrath broke the record of West Indies fast bowler Courtney Walsh. Title: Another gem in McGrath's career.
FLORES (tr)	Example	English: He built a WiFi door bell, he said. Turkish: WiFi ile çalışan bir kapı zili yaptığını söyledi.

Figure 6: A single in-context example/template from each respective benchmark. Blue indicates the continuation.

Model	**FLORES**	**XLSUM**	**MLAMA**	**Sentiment**	**XStoryCloze**	**XNLI**	**Average**
Phi-2	0.30	3.48	38.90	54.90	52.22	34.66	30.74
Gemma-2B	15.52	10.40	44.47	92.20	62.14	44.47	44.87
XGLM-7.5B	19.35	8.55	43.21	82.40	61.22	40.72	42.58
Bloom-7.1B	21.32	9.48	47.21	94.00	64.99	41.65	46.44
Gemma-7B	32.26	14.60	50.05	97.20	70.62	41.65	51.06
Mistral-7B	5.35	11.67	42.50	92.50	60.36	41.02	42.23
LLaMA-2-7B	8.17	10.74	38.92	89.30	57.45	36.63	40.20
Phi-2 (CPT)	0.96	4.68	39.84	56.00	53.28	33.57	31.39
Gemma-2B (CPT)	19.58	9.69	43.21	79.90	61.35	37.23	41.83
XGLM-7.5B (CPT)	20.61	6.64	43.86	69.40	59.56	40.44	40.09
Bloom-7.1B (CPT)	20.18	10.36	48.54	89.50	64.26	40.36	45.53
Gemma-7B (CPT)	31.75	12.12	49.24	98.20	69.69	39.04	50.01
Mistral-7B (CPT)	26.21	14.06	47.41	96.20	69.29	40.56	48.96
LLaMA-2-7B (CPT)	21.99	10.90	45.02	92.40	64.06	39.36	45.62
Phi-2 (ΔV=10K, CPT)	4.01	8.05	40.84	66.30	58.70	35.10	35.50
Gemma-2B (ΔV=10K, CPT)	21.85	9.86	43.42	92.80	63.53	40.16	45.27
XGLM-7.5B (ΔV=10K, CPT)	16.81	5.63	43.79	70.90	61.02	40.64	39.80
Bloom-7.1B (ΔV=10K, CPT)	25.43	11.18	48.02	86.50	64.20	39.36	45.78
Gemma-7B (ΔV=10K, CPT)	30.24	12.23	45.86	96.40	70.62	41.00	49.39
Mistral-7B (ΔV=10K, CPT)	25.21	12.14	43.42	81.90	65.32	41.97	44.99
LLaMA-2-7B (ΔV=10K, CPT)	25.24	11.99	44.14	92.10	65.65	41.04	46.69

Table 12: Adapted LLM's performance on Hindi, with continued fine-tuning on 100K examples.

Model	FLORES	XLSUM	MLAMA	Sentiment	XCOPA	Average
Phi-2	0.02	0.46	34.43	49.30	48.20	26.48
Gemma-2B	5.65	14.58	39.71	90.10	48.80	39.77
XGLM-7.5B	10.17	4.36	37.19	76.10	51.60	35.88
Bloom-7.1B	8.47	11.38	41.41	64.70	54.80	36.15
Gemma-7B	18.08	14.51	44.65	96.70	56.20	46.03
Mistral-7B	1.08	5.99	35.11	75.60	47.60	33.08
LLaMA-2-7B	1.54	3.95	36.81	62.10	48.60	30.60
Phi-2 (CPT)	0.02	0.37	35.64	49.50	45.40	26.19
Gemma-2B (CPT)	6.88	14.88	38.45	82.40	52.80	39.08
XGLM-7.5B (CPT)	8.90	7.11	39.71	61.60	53.00	34.06
Bloom-7.1B (CPT)	10.27	14.53	44.23	52.20	58.40	35.93
Gemma-7B (CPT)	19.72	13.67	43.64	97.20	63.80	47.60
Mistral-7B (CPT)	14.64	9.39	38.49	89.60	58.80	42.18
LLaMA-2-7B (CPT)	8.81	7.41	38.72	82.00	53.60	38.11
Phi-2 (ΔV=10K, CPT)	0.25	10.64	36.91	53.60	49.60	30.20
Gemma-2B (ΔV=10K, CPT)	8.92	12.91	38.70	91.10	53.40	41.01
XGLM-7.5B (ΔV=10K, CPT)	8.01	6.14	39.73	68.80	51.60	34.86
Bloom-7.1B (ΔV=10K, CPT)	12.33	14.69	44.57	51.00	57.20	35.96
Gemma-7B (ΔV=10K, CPT)	17.28	16.28	42.34	97.10	58.20	46.24
Mistral-7B (ΔV=10K, CPT)	12.66	17.34	39.21	91.80	58.60	43.92
LLaMA-2-7B (ΔV=10K, CPT)	11.32	16.61	42.12	93.10	55.00	43.63

Table 13: Adapted LLM's performance on Tamil, with continued fine-tuning on 100K examples.