One Tokenizer To Rule Them All: Emergent Language Plasticity via Multilingual Tokenizers

Diana Abagyan[1], **Alejandro R. Salamanca**[1], **Andres Felipe Cruz-Salinas**[2], **Kris Cao**[2], **Hangyu Lin**[2], **Acyr Locatelli**[2], **Marzieh Fadaee**[1], **Ahmet Üstün**[1,2], **Sara Hooker**[1,†]
[1]Cohere Labs [2]Cohere
diana@cohere.com

Abstract

Pretraining massively multilingual Large Language Models (LLMs) for many languages at once is challenging due to limited model capacity, scarce high-quality data, and compute constraints. Moreover, the lack of language coverage in the tokenizer makes it harder to address the gap for new languages purely at the post-training stage. In this work, we study what relatively cheap interventions early on in training improve "language plasticity", or adaptation capabilities of the model post-training to new languages. We focus on tokenizer design and propose using a *universal* tokenizer that is trained for more languages than the primary pretraining languages to enable efficient adaptation in expanding language coverage after pretraining. Our systematic experiments across diverse groups of languages and different training strategies show that a universal tokenizer enables significantly higher language adaptation, with up to 20.2% increase in win rates compared to tokenizers specific to pretraining languages. Furthermore, a universal tokenizer also leads to better plasticity towards languages that are completely unseen in the tokenizer and pretraining, by up to 5% win rate gain. We achieve this adaptation to an expanded set of languages with minimal compromise in performance on the majority of languages included in pre-training.

1 Introduction

There are only a handful of research labs with enough compute resources and expertise to train large AI systems at scale (Maslej et al., 2025; Hooker, 2024). Most researchers and practitioners are forced to choose among available pretrained models for downstream tasks, even if they are not tailored to their use cases. Nowhere is this tension more evident than in the multilingual setting (Joshi et al., 2020; Singh et al., 2024; Üstün et al., 2024), where limited investment in multilingual support in pretraining often results in significant gaps in language coverage in state-of-the-art LLMs (Holtermann et al., 2024).

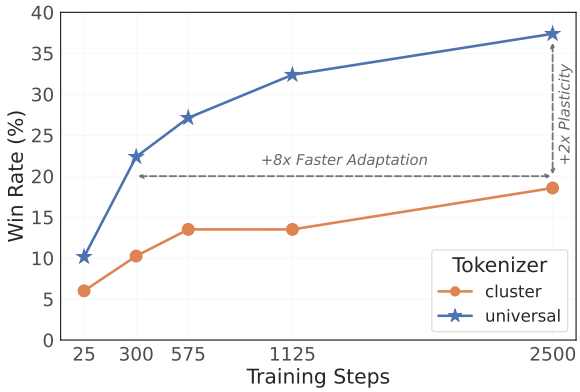


Figure 1: UNIVERSAL tokenizer exhibits **+2x higher plasticity** with **+8x faster adaptation** compared to the cluster-specific baseline tokenizer.

This imbalance in language coverage has created a growing divide in the cost of use for particular language users as marginalized languages require more tokens and incur higher latency for generations (Ji et al., 2023; Cui et al., 2024; Ahia et al., 2023), restricting speakers of low-performing languages to lower quality technology (Held et al., 2023; Durmus et al., 2024; Nicholas and Bhatia, 2023; Ojo et al., 2025). Further compounding these issues, once a model is pretrained, it is hard to steer towards new behavior using post-training alone (Wang et al., 2025). Unless the tokenizer has been calibrated to a new language during training, it often requires far more significant amount of data and intricate optimization steps (Muller et al., 2021).

Multilingual plasticity represents the capability of the language model to quickly adapt to lingual distribution shifts to the downstream target, which in our case, involves a new set of focus languages (Chen et al., 2023). Given that pretraining requires the bulk of compute and cost resources, any inter-

[†]Now at Adaption Labs.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 3118–3136

vention made at this stage that improves the *plasticity* for downstream developers and researchers is beneficial.

In this work, we investigate minimal and efficient pretraining interventions to reduce later adaptation costs. In particular, we identify tokenization as an area with relatively low cost of intervention, but potential for large downstream gains. We ask: *Can we leverage tokenizers with broad language coverage to improve the plasticity of LLMs without hurting pretraining performance?*

We hypothesize that a universal tokenizer that is trained on more languages than the primary pretraining languages, introduced from the start of pretraining, enables quick and effective interventions for adapting a model to new languages. This significantly diverges from the previous work that focuses on techniques such as vocabulary extension (Wang et al., 2020) or retraining the embedding layer (Artetxe et al., 2020) after pretraining. These techniques are more costly, needing more resources like training budget, with varying degrees of success across languages (Limisiewicz et al., 2023; Sharthak et al., 2025; Nag et al., 2025).

We systematically investigate the impact of multilingual tokenizers through an exhaustive set of ablations at pretraining scale, which requires a significant investment of resources. We vary tokenizer, language subsets, and adaptation strategies across 69 languages, and we find the following results:

1. The **UNIVERSAL** tokenizer significantly improves adaptation to new (expanded) languages, achieving an average of 19% higher win rate in continued pretraining experiments, compared to the baseline tokenizers specialized to pretraining languages. In addition to achieving higher adaptation, the **UNIVERSAL** tokenizer exhibits almost the same performance on primary languages with no more than 2% difference in downstream evaluation against the baseline tokenizer.

2. For targeted adaptation where the new languages are the only focus, the **UNIVERSAL** tokenizer achieves an average of 14.6% improvement over the baseline tokenizer for the expanded languages subset. Furthermore, for adapting to fully **unseen** languages, not included both in the tokenizer and pretraining (the most extreme case of adaptation), UNIVERSAL tokenizer outperforms the baseline by up to 5% gain in win rates across 7 heavily under-resourced languages.

3. We find that the UNIVERSAL tokenizer enables more than 8x faster adaptation performance, requiring much less additional training, and therefore minimal costs. We believe that this dramatically benefits practitioners who want to extend the language coverage of a pretraining model with minimal intervention.

2 Methodology and Experimental Setup

2.1 Methodology and Core Ablations

Language Coverage and Model Variants. Our experiments include 62 typologically and lexicographically diverse languages, broken up into three geographically motivated clusters: (1) European languages, (2) Asian languages, and (3) Middle-Eastern and Indic languages (referred to as ME-Indic throughout the paper). For each geo-cluster, we pretrain language models primarily on the languages within that cluster (referred to as **primary** subset) and we use the remaining languages (referred to as **expanded** subset) as reference points for plasticity adaptation experiments. For example, for the European cluster, the primary subset consists of languages such as Spanish, Russian, and Portuguese, and the expanded subset includes the 10 languages outside of the dominant training data for that cluster. In addition, we also consider 7 **fully unseen** languages, such as Sinhala and Kazakh, which were not present in the tokenizer or base model training data. The full language list with clusters is provided in Appendix E.

Adaptation Strategies. One of our goals is to introduce highly plastic and adaptable model properties. A practitioner may choose to conduct language adaptation using a variety of different training strategies, influenced by what data they may have available. Ideally, the choices we make for the tokenizer allow for improved plasticity given any approach taken after pretraining. Hence, we evaluate our interventions under various different adaptation strategies, which include continued pretraining with data in both **primary and expanded** language subsets, targeted adaptation for **expanded** languages, and targeted adaptation for **fully unseen** languages. We briefly describe both these strategies and experimental details below:

- **Continued pretraining with data from pri-**

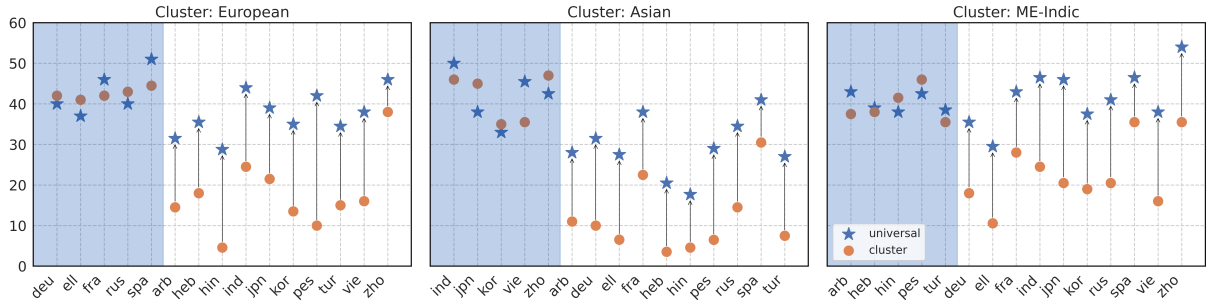


Figure 2: Win rates for models trained with the UNIVERSAL and CLUSTER tokenizers against Dolly generations (Singh et al., 2024). The shaded blue portion represents a subset of primary languages, and the white represents expanded languages. On average, UNIVERSAL tokenizer achieves an increase of 18.9% win rate on expanded subset languages, and +0.3% on primary languages compared to the CLUSTER tokenizers across clusters.

mary and expanded languages: The objective for this strategy is to increase language coverage of the model, so that it supports both primary and expanded languages. Half of the training mix consists of an even distribution of all languages in the instruction finetuning data, and the other half is a standard cooldown mix with high-quality datasets (See § 2.2). This imparts instruction-following abilities to our base models, and also allows for evaluation on both the primary and expanded languages.

- **Targeted adaptation (expanded languages):** In these experiments, we explore a targeted language adaptation through supervised finetuning. The post-training data solely consists of instruction-style data in the expanded language subsets for each cluster model. This allows us to isolate the effect of introducing new languages that are not focused on during pretraining but represented in the tokenizer.

- **Targeted adaptation (fully unseen languages):** In the final set of experiments, we explore the most extreme setting of targeted adaptation to fully unseen languages that are not seen in the tokenizer or pretraining. In this setting, we consider the availability of the data in one language only for each experiment, thus, we fine-tune the base model on one language at a time. This ablation enables evaluating our approach for adaptation under a heavily under-resourced scenario.

Tokenizer Variants. We train a massively multilingual tokenizer using data from all 62 languages, as well as cluster-specific tokenizers that represent only the primary language subsets. Throughout the paper, we refer to these tokenizers as **UNIVERSAL** and **CLUSTER** tokenizers, respectively. We include more details about the tokenizer training in Section 2.3.

2.2 Experimental Set-up

Pretraining datasets. Models are pretrained with a mixture of English, code, and multilingual corpora, where data weights are distributed as 55%, 15%, and 30%, respectively. Upweighting English in multilingual training is a common practice, due to higher task coverage and quality, which is crucial for cross-lingual transfer (Dash et al., 2025; Singh et al., 2024; Ravisankar et al., 2025). We also include code data as it has become a standard part of the training recipe for natural language models, and has been found to boost performance on other tasks when included in pretraining (MA et al.; Aryabumi et al., 2024). We use a large corpus of data from a variety of public and proprietary sources. For models we pretrain with the UNIVERSAL tokenizer, we reallocate 5% of the training mixture from English data and uniformly distribute it among all the expanded languages, to avoid undertrained tokens in the vocabulary (Land and Bartolo, 2024). However, in Section 5.4, we ablate this percentage and show that even when no expanded language subset data is included in pretraining, the UNIVERSAL tokenizer significantly improves multilingual plasticity.

Cooldown and instruct datasets. For continued pretraining, we use a cooldown mixture that involves upweighting higher quality datasets, comprised of text, math, code, and instruct-style data (Aryabumi et al., 2024). Cooldown has been proven to improve performance on downstream tasks, including instruction-following (Parmar et al., 2024; Team et al., 2025). We include

a high-quality mix of proprietary and open data, much of which was created by following a multilingual data arbitrage strategy (Odumakinde et al., 2024), covering 100,000 prompt-completion pairs in 23 languages. Finally, for experiments on fully unseen languages, we emulate a realistic data-constrained regime, and use only 14,800 instructions per language from the translated Dolly training set from Aya Collection (Singh et al., 2024).

Training details. For our experiments, as standard for most LLMs, we use the Transformer-based decoder-only architecture (Vaswani et al., 2017; Radford and Narasimhan, 2018). Our architecture includes key optimizations such as Parallel Attention Blocks (Chowdhery et al., 2023), Grouped Query Attention (Ainslie et al., 2023), SwiGLU activation function (Shazeer, 2020), and Rotary Positional Embeddings (Su et al., 2024). Additional training and infrastructure details are provided in Appendix C.

2.3 Tokenizer Training

All tokenizers are trained using the Byte Pair Encoding algorithm (Sennrich et al., 2016). Additional implementation details about the tokenizer training is given in Appendix A. We use a vocabulary size of 250k tokens in our main experiments, although we experiment with various sizes in § 5.3.

$$w_i = \frac{w_i^d . w_i^b}{\sum_n w_n^d . w_n^b} \quad (1)$$

Language weighting. In addition to varying the coverage of the tokenizer on either UNIVERSAL or CLUSTER language coverage, we also invest in a methodology that adjusts the weighting based upon availability of data. In contrast to traditional approaches which sample uniformly across all data and end up dominated by most frequent languages, we consider two factors: (1) natural distribution of the data available across languages, and (2) language buckets formed by languages that share the same family and script (which are more likely to share tokens). Within each language bucket, we use uniform weighting across languages. Concretely, for a language i, where w_i^d and w_i^b denote weights for data distribution and language bucket, respectively, we compute the language weights in the tokenizer data mixture following Equation 1.

This way, we balance natural data distribution (skewed through the high-resource languages) with language bucketing in a principled manner, ensuring that there is equitable representation for diverse scripts and lower-resourced languages. Our pretraining experiments (Section 3, Appendix B) show that our specialized weighting combining language bucketing with size-proportional data distribution enables better compression ratios than uniform weighting and achieves better downstream performance.

2.4 Evaluation

Open-ended evaluation. Goldman et al. (2024) find that generative tasks are more informative than classification in evaluating tokenizers, likely due to the number of generation steps. Following Üstün et al. (2024), the quality of generations is assessed using LLM-as-a-Judge win rates, where original generations are used as the reference answer. We use the `dolly_human_edited` and the `dolly_machine_translated` splits of the Aya Evaluation Dataset (Singh et al., 2024) as test data for this task, which are formed by translating 200 held-out examples from the Dolly-15k (Conover et al., 2023). We use 15 adaptation languages for open-ended evaluation, listed in Appendix E.

Prior work has shown that LLMs as evaluators are reasonable proxies and aligned with human preferences also in multilingual settings (Üstün et al., 2024; Singh et al., 2025; Dang et al., 2024; Kreutzer et al., 2025). We use Command-A (Cohere et al., 2025) as the judge model, given its reported strength as the best open-weights judge on multilingual setting, scoring closely to GPT4o (Gureja et al., 2024; Pombal et al., 2025). The full judge prompt is included in Appendix D.2.

Task-specific performance. We use two task-specific benchmarks for multilingual evaluation: Belebele (Bandarkar et al., 2024) is a multiple-choice question machine-reading comprehension (MRC) dataset representing 122 language variants. Multilingual MMLU (M-MMLU) (Dac Lai et al., 2023) is a machine-translated version of the original MMLU dataset (Hendrycks et al., 2021) that contains questions ranging in topic from STEM to humanities.

English-only evaluation. Additionally, we also evaluate models on 11 English-only natural language inference and commonsense reasoning benchmarks: ARC-C and ARC-E (Chollet, 2019), BoolQ (Clark et al., 2019), CommonsenseQA (Talmor et al., 2019), Hellaswag (Zellers et al., 2019), MMLU (Hendrycks et al., 2021), OpenBookQA (Mihaylov et al., 2018), PIQA (Bisk et al., 2020), SIQA (Sap et al., 2019), TruthfulQA (Lin et al.,

Cluster	Tokenizer	Belebele	M-MMLU	EN Tasks
		PRIMARY LANGUAGES		
European	CLUSTER	41.4	31.1	48.5
	UNIVERSAL	41.9	30.9	48.4
Asian	CLUSTER	38.2	29.6	48.2
	UNIVERSAL	38.1	28.9	48.1
Middle East & Indic	CLUSTER	38.1	29.2	49.1
	UNIVERSAL	36.5	28.6	48.2

Table 1: Comparison of CLUSTER vs. UNIVERSAL tokenizers during the **pretraining** on the primary languages across three regional clusters. Performance with UNIVERSAL tokenizer is comparable to CLUSTER tokenizers across all the geo-cluster models.

2022), and WinoGrande (Sakaguchi et al., 2019).

We include task-specific evaluations (both multilingual and English-only) to understand the relative merit of different design choices. Typically pretrained models do not perform well at downstream tasks at this point in training, as the models have not yet been optimized for instruction following (Wang et al., 2022; Üstün et al., 2024; Aakanksha et al., 2024), or aligned using reinforcement learning (Ahmadian et al., 2024; Dang et al., 2024). Hence, we do not expect state-of-the-art performance, but rather evaluate the relative signal of different variants.

3 Results on Pretraining Performance

In this section, we first benchmark the performance of our pretrained models, to ensure that using a UNIVERSAL tokenizer doesn't cause degradations in performance, as may be expected when using a tokenizer optimized for a broader set of languages than the primary set.

UNIVERSAL tokenizer does not compromise performance on primary languages. As seen in Table 1, we find that our expanded UNIVERSAL tokenizer is remarkably competitive against CLUSTER across the geo-cluster models. The difference in pretraining performance is less than at most 1% average accuracy in English tasks. The highest performance difference between UNIVERSAL and CLUSTER tokenizers for multilingual tasks is only 1.6% average accuracy on Belebele for ME-Indic cluster (38.1% vs 36.5 %). Overall, we observe minimal trade-offs in performance on primary cluster languages switching to UNIVERSAL tokenizer.

In fact, we observe that the UNIVERSAL tokenizer leads to a slight increase on average on Belebele for Euro cluster (41.9 vs 41.4) and achieves much closer performance for Asian cluster (38.1 vs 38.2). As additional validation, Figure 8 in Appendix D.1 shows the progression of average Belebele performance for both tokenizers for Euro cluster models during pretraining. The UNIVERSAL tokenizer achieves approximately similar performance throughout the whole pretraining, also suggesting the same trend in a longer pretraining run. Overall, these results showcase that using a UNIVERSAL tokenizer doesn't spell any significant performance degradation in pretraining for the primary languages.

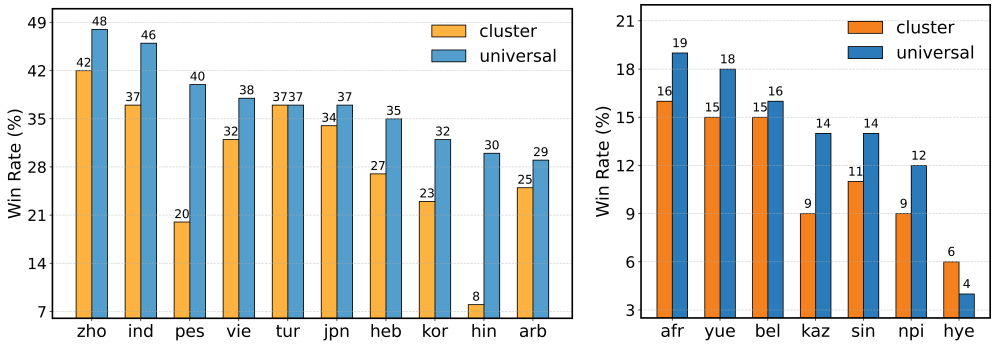
Balanced language weighting with language buckets for tokenizer training leads to better pretraining performance. As described in Section 2.3 on tokenizer training, we weight the languages using buckets formed by script and language family, balanced against data availability. In order to motivate this weighting scheme, we compare pretraining performance of UNIVERSAL tokenizer against a baseline tokenizer (UNIFORM), where all languages are uniformly weighted except English. We conduct this ablation in the Euro cluster, where the number of primary languages is the highest. In tokenizer training, we use all the languages (62 languages; primary and expanded subsets) and only vary the language weighting. As shown in Table 3, UNIVERSAL tokenizer with balanced weighting using language buckets outperforms UNIFORM weighting in 21 European languages out of 27, with a relative gain of 2.2% (41.9 vs 41.0) on average. Further validating pretraining results, we provide the comparison for compression performance between these two tokenizers in Appendix B, where the results show better overall compression in UNIVERSAL tokenizer.

4 Results on Enhanced Multilingual Plasticity

4.1 Benefits of Plasticity in Continued Pretraining

In this section, we ask: *Does varying the approach for the tokenizer lead to plasticity benefits after continued pretraining on both primary and expanded languages?*

Models trained with the UNIVERSAL tokenizer demonstrate significantly higher win rates on the EXPANDED SUBSET. Figure 2 and Table 2a show results of evaluation across the Euro, Asian, and ME-Indic clusters, with 5 languages belonging to each PRIMARY LANGUAGE SUBSET and 10



(a) **Expanded** languages (b) Fully **unseen** languages.

Figure 3: Language-specific results after **targeted adaptation** through SFT for the Euro cluster model to **expanded** languages (a) and fully **unseen** languages (b). UNIVERSAL tokenizer outperforms the CLUSTER tokenizer in both language subsets, where the relative gains over the cluster-specific tokenizer go up to 22% on expanded languages (Hindi), and 5% on unseen languages (Kazakh).

in the EXPANDED SUBSET belonging to the other two clusters. We see that the UNIVERSAL tokenizer achieves an average gain of 18.9% in win rates across all three geo-cluster models on the expanded subsets, compared to models trained with the CLUSTER tokenizer. Improvement is consistent across clusters, where we find +19.9%, +17.8%, and +18.9% increase in win rate for Euro, Asian, and ME-Indic cluster models, respectively. Among all the expanded languages, Persian (+25.8%), Hindi (+23.3%), and Vietnamese (+22.0%) show the highest benefit from the UNIVERSAL tokenizer in the Euro, Asian, and ME-Indic clusters, respectively.

UNIVERSAL tokenizer preserves performance on all geo-clusters. While the UNIVERSAL tokenizer provides significant gains on the expanded languages, in Figure 2, we observe that the performance on primary languages is nearly the same across both tokenizers in all three clusters. There is only a 0.3% win rate difference across all clusters for primary languages when comparing tokenizers, where UNIVERSAL tokenizer even leads to a slight increase over the CLUSTER tokenizer in the European, Asian, and ME-Indic models by 0.3%, 0.1%, and 0.5%, respectively. This is beneficial, as it suggests no trade-offs of improving plasticity for an expanded set of languages by using the UNIVERSAL tokenizer for the primary languages a provider is interested in when developing a model.

In machine translation, models trained with the UNIVERSAL tokenizer outperform CLUSTER baseline across all the languages. To extend evaluation beyond LLM-as-a-judge, we evaluate our cluster models trained with UNIVERSAL and CLUSTER tokenizers after continued pretraining on Flores-200 (Team et al., 2022), for both primary and extended languages. We find that the advantage of the UNIVERSAL tokenizer is exhibited in this evaluation as well, with 5.3 average increase in BLEU score on the expanded languages, and even an improvement in the primary languages of 2.2 BLEU.

4.2 Benefits of Plasticity in Targeted Adaptation

An experimental setting of great interest is the more realistic scenario where a downstream developer only has access to data in the EXPANDED languages. To mimic this scenario, we evaluate the impact of our interventions when only supervised fine-tuning only on the expanded language subset is feasible.

UNIVERSAL tokenizer outperforms CLUSTER tokenizers by high margins in targeted adaptation for expanded language set. Table 2b shows average win rates of UNIVERSAL and CLUSTER tokenizers for each geo-cluster. UNIVERSAL tokenizer achieves 10.2%, 15.7%, and 17.8% relative win rate gains over the CLUSTER-specific tokenizers for Euro, Asian, and ME-Indic clusters, respectively. In Figure 3a, we also plot individual language gains for the Euro cluster. UNIVERSAL consistently enables higher plasticity than CLUSTER tokenizer where the relative gains go up to 22.0% and 20.0% in Hindi and Farsi, respectively.

UNIVERSAL tokenizer also provides large gains in targeted adaptation for fully unseen language set. In the most extreme setting, we evaluate the benefits of our tokenizer intervention for adaptation to languages that are **fully unseen** in both

Cluster	Tokenizer	Dolly Win Rates (%) PRIMARY	Dolly Win Rates (%) EXPANDED	Flores-200 (BLEU) PRIMARY	Flores-200 (BLEU) EXPANDED
European	CLUSTER	42.8	17.6	23.5	8.7
	UNIVERSAL	42.8	37.4 (+19.9)	24.3	13.4 (+4.7)
Asian	CLUSTER	41.7	11.7	11.3	7.7
	UNIVERSAL	41.8	29.5 (+17.8)	14.2	15.7 (+8.0)
Middle East & Indic	CLUSTER	39.7	22.8	12.3	13.2
	UNIVERSAL	40.2	41.8 (+18.9)	15.1	16.5 (+3.3)

(a) **Continued pretraining**

	Dolly Win Rates (%) CLUSTER	Dolly Win Rates (%) UNIVERSAL
European	27.2	37.4 (+10.2)
Asian	18.8	34.3 (+15.7)
Middle East & Indic	23.31	41.1 (+17.8)

(b) **Targeted adaptation**

Table 2: (a) The UNIVERSAL tokenizer matches CLUSTER performance on primary languages and shows large gains (up to 19.9% winrates in Dolly and up to 8 BLEU score in Flores) on average of expanded language subsets across all clusters. (b) Win rates on expanded languages after **targeted adaptation**. The UNIVERSAL tokenizer shows better performance (up to 17.8%) across all clusters over the baseline CLUSTER tokenizer.

tokenizer and pretraining. It is critical that all these languages are extremely under-resourced, and this adaptation is performed in a low-data environment, as this is representative of the constraints faced by developers in these languages. Figure 3b shows results on supervised fine-tuning experiments on 7 unseen languages. We find that UNIVERSAL tokenizer enables improvements over the CLUSTER tokenizer on the unseen languages with an average gain of 2.0% in win rates, where it goes up to 5.0% in Nepali. To cover all bases, in Appendix D.1.3 we ensure this advantage is preserved when adapting to an unseen language in a higher-data setting.

Given that the downstream performance is generally lower for these languages due to their absence in the tokenizer and pretraining, and also constraints on available data (only 15k per language), we find this to be a promising direction of future research and another reason to invest in more flexible tokenizer design.

5 Key Discussions

5.1 Comparison with cross-lingual vocabulary adaptation

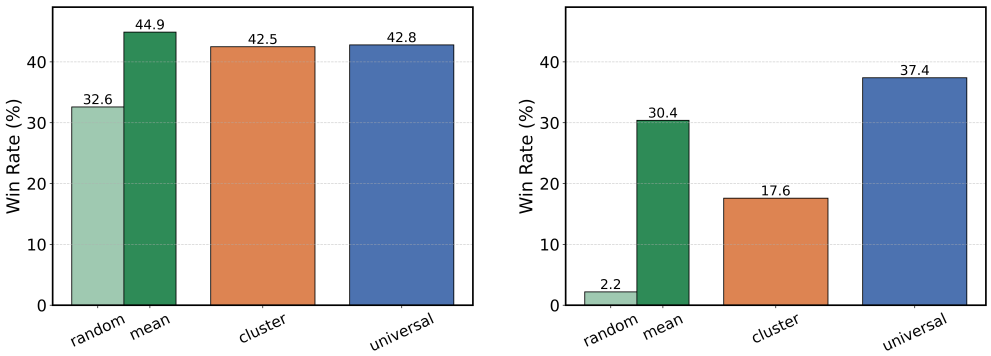
Cross-lingual vocabulary adaptation (CVA) (Yamaguchi et al., 2024b) aims to adapt the existing tokenizer and hence the token embeddings to the new languages through expansion or replacement after pretraining, and is a common approach for language adaptation. A more detailed overview of CVA can be found in Section 6. In this ablation, we ask: *how does the* UNIVERSAL *tokenizer compare with CVA for adapting new languages?*

For our simple CVA baseline, the tokenizer of the pretrained Euro cluster model that is trained with the CLUSTER tokenizer is replaced with the UNIVERSAL tokenizer. We choose to compare against vocabulary replacement, rather than expansion, so that vocabulary size doesn't play a role, and to isolate the effects of pretraining versus post-training intervention. Token embeddings that are shared between CLUSTER and UNIVERSAL tokenizers are preserved, and new tokens are either randomly initialized by sampling from a normal distribution (`random`), or with the average of the shared embeddings (`mean`). After vocabulary replacement and token initialization, we follow the same continued pretraining described in Section 2.1.

The results for this ablation are given in Figure 4. We find that when randomly initializing the new tokens, CVA (tokenizer replacement) fails to achieve comparable performance even against the CLUSTER tokenizer, and significantly lags behind the UNIVERSAL tokenizer by 15.4% and 35.2% win rates for primary and expanded languages respectively. Notably, initializing the new tokens with the mean of the shared vocabulary (`mean` in Figure 4), outperforms random initialization. While tokenizer replacement (`mean`) is an improvement over the unadapted CLUSTER tokenizer by 12.8% relative increase in win rates for expanded languages, our UNIVERSAL tokenizer leads to better adaptation performance by 7% difference in average win rate (37.4% vs 30.4%). Interestingly, CVA (`mean`) achieves slightly higher performance in the primary languages by 2.1% average win rate. Overall, these results show that it is more effective to use a UNIVERSAL tokenizer from the start, rather than substituting it in after pretraining.

5.2 Adaptation Efficiency with the Universal Tokenizer

In this ablation, we evaluate how much faster adaptation takes place with the UNIVERSAL tokenizer.



(a) **Primary** languages.

(b) **Expanded** languages

Figure 4: Win rates after continued pretraining, comparing Crosslingual Vocabulary Adaptation (CVA) through tokenizer replacement. The UNIVERSAL tokenizer significantly outcompetes both CVA approaches on expanded languages.

Faster adaptation means much fewer resources necessary, namely cost, which is of great interest to practitioners who want to adapt an LLM to expanded language coverage.

To evaluate the adaptation speed, in comparison to CLUSTER tokenizer, we evaluated the intermediate checkpoints on the expanded languages during continued pretraining for Euro cluster models. Figure 1 shows average win rates on 10 expanded languages. As seen in the plot, in only 300 steps UNIVERSAL tokenizer reaches the level of performance that CLUSTER achieves at 2500 steps, showing +8x faster adaptation. Given that 300 steps correspond to nearly 150K samples (compared to 1.3M at 2500 steps), UNIVERSAL tokenizer requires also much less data to achieve the same performance with the end performance of the baseline, confirming the effectiveness of our proposal.

5.3 Necessity of Large Vocabulary Size

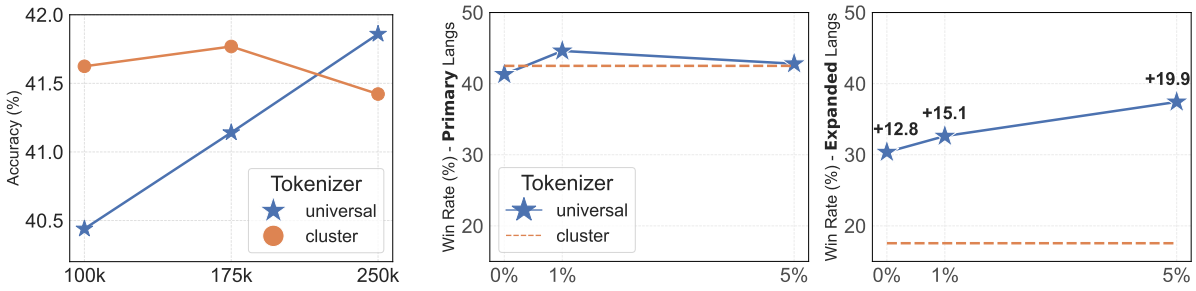
In the previous sections, we establish greater performance in multilingual plasticity with the UNIVERSAL compared to CLUSTER tokenizer. In this ablation, we ask: *What is the required vocabulary size for the* UNIVERSAL *tokenizer to avoid performance degradation on primary pretraining languages?*

To determine the optimal vocabulary size, we run additional pretraining experiments where we vary the vocabulary size from 100,000 tokens to 250,000 tokens while adjusting the model parameters so that the total number of trainable parameters remains constant. We evaluate the performance for the primary pretraining languages on Belebele. Results are shown in Figure 5a. The models trained with CLUSTER tokenizers don't vary much in performance, and surpass the UNIVERSAL tokenizer at small vocabulary sizes (100k and 175k). However, the UNIVERSAL tokenizer scales performance with the vocabulary size, and overtakes the CLUSTER tokenizer at 250k vocabulary size. Our findings are consistent with previous work that shows the benefits of large vocabularies (Tao et al., 2024; Huang et al., 2025), and suggests investment in universal tokenizers require a reallocation of weights to ensure a proper vocabulary budget. Based on this ablation, we use the vocabulary size of 250k in our main pretraining runs.

5.4 Presence of Expanded Language Subset in Pretraining

In the large and often noisy datasets used to train LLMs, there is often language contamination (Blevins and Zettlemoyer, 2022). Therefore, it can be difficult to claim a language is truly "new". In our final ablation, to test the robustness of our claims of plasticity under different assumptions of multilingual data presence, we evaluate 0%, 1%, and 5% proportion of expanded languages in pretraining for the European cluster.

Figure 5b shows that even in the most conservative case of 0% multilingual percentage for the new languages (the expanded subset), the UNIVERSAL tokenizer exhibits 12.8% gain in win rate as compared to the CLUSTER tokenizer. Notably, increasing this percentage up to 5% does not hurt performance in primary pretraining languages, but increases adaptation performance on the expanded languages from 12.8% to 19.8% win rates.



(a) Effect of vocabulary size.

(b) Effect of the expanded language data in pretraining

Figure 5: (a) UNIVERSAL tokenizer requires a larger vocabulary size to achieve the same (or better) pretraining performance with the CLUSTER tokenizers on **primary** languages, as evaluated in Belebele. (b) UNIVERSAL tokenizer exhibits significantly higher adaptation gains over the CLUSTER tokenizer even when there is no data from the expanded languages added in pretraining.

6 Related Work

There are a number of ways that language adaption of pretrained language models is commonly approached. In terms of additional training, continued pretraining (CPT) and supervised fine-tuning (SFT) are the most common methods. Continued pretraining involves extended training on the adaptation language corpora (Han and Eisenstein, 2019; Muller et al., 2021), but requires a significant amount of data, which may not be accessible for lower-resourced languages. Supervised fine-tuning is also a standard approach (Kumar et al., 2022; Adelani et al., 2021; Cahyawijaya et al., 2021), requiring less data than CPT, but possibly leading to catastrophic forgetting of capabilities from pretraining (Rolnick et al., 2019; Chaudhry et al., 2019). In particular, instruction fine-tuning is popular in order to impart instruction-following capabilities as well (Gala et al., 2024). Claims as to the advantages of one over the other are mixed- (Ebrahimi and Kann, 2021) find that in their setup, CPT is more effective than SFT, and Yong et al. (2023) find the opposite.

A primary challenge is unsupported scripts and languages in the tokenizer. Cross-lingual vocabulary adaptation (CVA) modifies the existing tokenizer to accommodate additional languages (Yamaguchi et al., 2024a), and requires continued pretraining in the target language to sufficiently adapt (Fujii et al., 2024). There are two common approaches to CVA – vocabulary expansion, where new tokens are added from the target and shared tokens are reused (Wang et al., 2020; Pfeiffer et al., 2021), or vocabulary replacement, where the vocabulary is entirely replaced. Embeddings corresponding to new tokens may be initialized randomly, using heuristics such as the average of some corresponding tokens in the original vocabulary (Minixhofer et al., 2022; Dobler and de Melo, 2023; Downey et al., 2023), or based on auxiliary models (Ostendorff and Rehm, 2023). Switching out the tokenizer is cumbersome; one possible method is by training a hypernetwork that maps vocab of the new tokenizer to existing embeddings (Minixhofer et al., 2025). This method requires continued training to close the performance gap, but even then doesn't surpass it. It has also been proposed to transliterate languages to Latin script to circumvent unsupported scripts (Muller et al., 2021), but this approach is limited by transliteration performance.

Conclusion

In this work, we explore what cheap interventions in pretraining can increase plasticity in downstream optimization stages. We conduct an extensive study involving different tokenization strategies, three language adaptation strategies involving different assumptions about data access across 70 different languages. We find that in all cases, a model trained using a UNIVERSAL tokenizer with broad language coverage is able to adapt to languages outside of the primary pretraining set far better, with average win rate improvements up to 20.2% in continued pretraining and 17.8% in targeted adaptation to expanded languages. Even in the challenging low-data setting of completely unseen languages, the UNIVERSAL tokenizer shows gains up to 5%. At the same time, there is negligible performance impact to the primary pretraining languages. Investing in a massively multilingual tokenizer up-front pays off in language adaptation down the line.

Limitations

Firstly, our experiments involve 70 languages covering a diverse range of languages and scripts, where we systematically investigate the impact of multilingual data on tokenizer training. Although we use a comprehensive list of languages, there are many more languages in the world, which requires attention of the research community. We hope our work encourages even broader language coverage in state-of-the-art language models.

Secondly, we focused on the BPE algorithm for tokenizer training, which is the most widely used method for language models. This choice was dictated by the high computational cost of each ablation, which required significant compute resources. However, we believe our findings on multilingual coverage would apply to the other tokenizers such as Unigram (Kudo, 2018), or byte or character-level tokenization (Xue et al., 2022; Clark et al., 2022). We leave this exploration to future research.

Finally, our experiments were conducted on 3.3B parameter models with a 100B token budget, which is an enormous undertaking for resources and compute costs. Given that our results hold for this scale, we anticipate they would also apply to larger models and token budgets, as supported by previous research (Biderman et al., 2023; Longpre et al., 2024; Aryabumi et al., 2024).

Acknowledgments

We thank Julia Kreutzer, John Dang, Roman Castagné, Aakanksha, and other colleagues at Cohere and Cohere Labs for their support and thoughtful feedback. We also thank Shayne Longpre for his feedback on the preprint.

References

Aakanksha, Arash Ahmadian, Seraphina Goldfarb-Tarrant, Beyza Ermis, Marzieh Fadaee, and Sara Hooker. 2024. Mix data or merge models? optimizing for diverse multi-task learning. *Preprint*, arXiv:2410.10801.

David Ifeoluwa Adelani, Jade Abbott, Graham Neubig, Daniel D'souza, Julia Kreutzer, Constantine Lignos, Chester Palen-Michel, Happy Buzaaba, Shruti Rijhwani, Sebastian Ruder, Stephen Mayhew, Israel Abebe Azime, Shamsuddeen H. Muhammad, Chris Chinenye Emezue, Joyce Nakatumba-Nabende, Perez Ogayo, Aremu Anuoluwapo, Catherine Gitau, Derguene Mbaye, and 42 others. 2021. Masakhaner: Named entity recognition for african languages. *Transactions of the Association for Computational Linguistics*, 9:1116–1131.

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.

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. 2024. Back to basics: Revisiting REINFORCE-style optimization for learning from human feedback in LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12248–12267, Bangkok, Thailand. Association for Computational Linguistics.

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebron, and Sumit Sanghai. 2023. GQA: Training generalized multi-query transformer models from multi-head checkpoints. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4895–4901, Singapore. Association for Computational Linguistics.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. On the cross-lingual transferability of monolingual representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4623–4637, Online. Association for Computational Linguistics.

Viraat Aryabumi, Yixuan Su, Raymond Ma, Adrien Morisot, Ivan Zhang, Acyr Locatelli, Marzieh Fadaee, Ahmet Üstün, and Sara Hooker. 2024. To code, or not to code? exploring impact of code in pre-training. *Preprint*, arXiv:2408.10914.

Lucas Bandarkar, Davis Liang, Benjamin Muller, Mikel Artetxe, Satya Narayan Shukla, Donald Husa, Naman Goyal, Abhinandan Krishnan, Luke Zettlemoyer, and Madian Khabsa. 2024. The belebele benchmark: a parallel reading comprehension dataset in 122 language variants. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 749–775, Bangkok, Thailand. Association for Computational Linguistics.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, and 1 others. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2020. Piqa: Reasoning about

physical commonsense in natural language. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*.

Terra Blevins and Luke Zettlemoyer. 2022. Language contamination helps explains the cross-lingual capabilities of English pretrained models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3563–3574, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Samuel Cahyawijaya, Genta Indra Winata, Bryan Wilie, Karissa Vincentio, Xiaohong Li, Adhiguna Kuncoro, Sebastian Ruder, Zhi Yuan Lim, Syafri Bahar, Masayu Khodra, Ayu Purwarianti, and Pascale Fung. 2021. IndoNLG: Benchmark and resources for evaluating Indonesian natural language generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8875–8898, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K. Dokania, Philip H. S. Torr, and Marc'Aurelio Ranzato. 2019. On tiny episodic memories in continual learning. *Preprint*, arXiv:1902.10486.

Yihong Chen, Kelly Marchisio, Roberta Raileanu, David Adelani, Pontus Lars Erik Saito Stenetorp, Sebastian Riedel, and Mikel Artetxe. 2023. Improving language plasticity via pretraining with active forgetting. In *Advances in Neural Information Processing Systems*, volume 36, pages 31543–31557. Curran Associates, Inc.

François Chollet. 2019. On the measure of intelligence. *Preprint*, arXiv:1911.01547.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, and 1 others. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. Boolq: Exploring the surprising difficulty of natural yes/no questions. In *NAACL*.

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

Team Cohere, Aakanksha, Arash Ahmadian, Marwan Ahmed, Jay Alammar, Yazeed Alnumay, Sophia Althammer, Arkady Arkhangorodsky, Viraat Aryabumi, Dennis Aumiller, Raphaël Avalos, Zahara Aviv, Sammie Bae, Saurabh Baji, Alexandre Barbet, Max Bartolo, Björn Bebensee, Neeral Beladia, Walter Beller-Morales, and 207 others. 2025. Command a: An enterprise-ready large language model. *Preprint*, arXiv:2504.00698.

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. 2023. Free dolly: Introducing the world's first truly open instruction-tuned llm.

Yiming Cui, Ziqing Yang, and Xin Yao. 2024. Efficient and effective text encoding for chinese llama and alpaca. *Preprint*, arXiv:2304.08177.

Viet Dac Lai, Chien Van Nguyen, Nghia Trung Ngo, Thuat Nguyen, Franck Dernoncourt, Ryan A Rossi, and Thien Huu Nguyen. 2023. Okapi: Instruction-tuned large language models in multiple languages with reinforcement learning from human feedback. *arXiv e-prints*, pages arXiv–2307.

John Dang, Shivalika Singh, Daniel D'souza, Arash Ahmadian, Alejandro Salamanca, Madeline Smith, Aidan Peppin, Sungjin Hong, Manoj Govindassamy, Terrence Zhao, Sandra Kublik, Meor Amer, Viraat Aryabumi, Jon Ander Campos, Yi-Chern Tan, Tom Kocmi, Florian Strub, Nathan Grinsztajn, Yannis Flet-Berliac, and 26 others. 2024. Aya expanse: Combining research breakthroughs for a new multilingual frontier. *Preprint*, arXiv:2412.04261.

Saurabh Dash, Yiyang Nan, John Dang, Arash Ahmadian, Shivalika Singh, Madeline Smith, Bharat Venkitesh, Vlad Shmyhlo, Viraat Aryabumi, Walter Beller-Morales, Jeremy Pekmez, Jason Ozuzu, Pierre Richemond, Acyr Locatelli, Nick Frosst, Phil Blunsom, Aidan Gomez, Ivan Zhang, Marzieh Fadaee, and 6 others. 2025. Aya vision: Advancing the frontier of multilingual multimodality. *Preprint*, arXiv:2505.08751.

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.

Esin Durmus, Karina Nguyen, Thomas I. Liao, Nicholas Schiefer, Amanda Askell, Anton Bakhtin, Carol Chen, Zac Hatfield-Dodds, Danny Hernandez, Nicholas Joseph, Liane Lovitt, Sam McCandlish, Orowa Sikder, Alex Tamkin, Janel Thamkul, Jared Kaplan, Jack Clark, and Deep Ganguli. 2024. Towards measuring the representation of subjective global opinions in language models. *Preprint*, arXiv:2306.16388.

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.

Kazuki Fujii, Taishi Nakamura, Mengsay Loem, Hiroki Iida, Masanari Ohi, Kakeru Hattori, Hirai Shota, Sakae Mizuki, Rio Yokota, and Naoaki Okazaki. 2024. Continual pre-training for cross-lingual llm adaptation: Enhancing japanese language capabilities. *ArXiv*, abs/2404.17790.

Philip Gage. 1994. A new algorithm for data compression. *The C Users Journal archive*, 12:23–38.

Jay Gala, Thanmay Jayakumar, Jaavid Aktar Husain, Aswanth Kumar M, Mohammed Safi Ur Rahman Khan, Diptesh Kanojia, Ratish Puduppully, Mitesh M. Khapra, Raj Dabre, Rudra Murthy, and Anoop Kunchukuttan. 2024. Airavata: Introducing hindi instruction-tuned llm. *Preprint*, arXiv:2401.15006.

Matthias Gallé. 2019. Investigating the effectiveness of BPE: The power of shorter sequences. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1375–1381, Hong Kong, China. Association for Computational Linguistics.

Omer Goldman, Avi Caciularu, Matan Eyal, Kris Cao, Idan Szpektor, and Reut Tsarfaty. 2024. Unpacking tokenization: Evaluating text compression and its correlation with model performance. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 2274–2286, Bangkok, Thailand. Association for Computational Linguistics.

Srishti Gureja, Lester James V Miranda, Shayekh Bin Islam, Rishabh Maheshwary, Drishti Sharma, Gusti Winata, Nathan Lambert, Sebastian Ruder, Sara Hooker, and Marzieh Fadaee. 2024. M-rewardbench: Evaluating reward models in multilingual settings. *arXiv preprint arXiv:2410.15522*.

Xiaochuang Han and Jacob Eisenstein. 2019. Unsupervised domain adaptation of contextualized embeddings for sequence labeling. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4238–4248, Hong Kong, China. Association for Computational Linguistics.

William Held, Camille Harris, Michael Best, and Diyi Yang. 2023. A material lens on coloniality in nlp. *Preprint*, arXiv:2311.08391.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Preprint*, arXiv:2009.03300.

Carolin Holtermann, Paul Röttger, Timm Dill, and Anne Lauscher. 2024. Evaluating the elementary multilingual capabilities of large language models with MultiQ. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 4476–4494, Bangkok, Thailand. Association for Computational Linguistics.

Sara Hooker. 2024. On the limitations of compute thresholds as a governance strategy. *ArXiv*, abs/2407.05694.

Hongzhi Huang, Defa Zhu, Banggu Wu, Yutao Zeng, Ya Wang, Qiyang Min, and Xun Zhou. 2025. Over-tokenized transformer: Vocabulary is generally worth scaling. *Preprint*, arXiv:2501.16975.

Yunjie Ji, Yan Gong, Yong Deng, Yiping Peng, Qiang Niu, Baochang Ma, and Xiangang Li. 2023. Towards better instruction following language models for chinese: Investigating the impact of training data and evaluation. *Preprint*, arXiv:2304.07854.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The state and fate of linguistic diversity and inclusion in the NLP world. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, Online. Association for Computational Linguistics.

Mohammed Safi Ur Rahman Khan, Priyam Mehta, Ananth Sankar, Umashankar Kumaravelan, Sumanth Doddapaneni, Suriyaprasaad B, Varun G, Sparsh Jain, Anoop Kunchukuttan, Pratyush Kumar, Raj Dabre, and Mitesh M. Khapra. 2024. IndicLLMSuite: A blueprint for creating pre-training and fine-tuning datasets for Indian languages. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15831–15879, Bangkok, Thailand. Association for Computational Linguistics.

Julia Kreutzer, Eleftheria Briakou, Sweta Agrawal, Marzieh Fadaee, and Kocmi Tom. 2025. D\'ej\a vu: Multilingual llm evaluation through the lens of machine translation evaluation. *arXiv preprint arXiv:2504.11829*.

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.

Aman Kumar, Himani Shrotriya, Prachi Sahu, Amogh Mishra, Raj Dabre, Ratish Puduppully, Anoop Kunchukuttan, Mitesh M. Khapra, and Pratyush Kumar. 2022. IndicNLG benchmark: Multilingual datasets for diverse NLG tasks in Indic languages. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5363–5394, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Sander Land and Max Bartolo. 2024. Fishing for magikarp: Automatically detecting under-trained tokens in large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 11631–11646, Miami, Florida, USA. Association for Computational Linguistics.

Tomasz Limisiewicz, Jiří Balhar, and David Mareček. 2023. Tokenization impacts multilingual language modeling: Assessing vocabulary allocation and overlap across languages. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5661–5681, Toronto, Canada. Association for Computational Linguistics.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland. Association for Computational Linguistics.

Shayne Longpre, Gregory Yauney, Emily Reif, Katherine Lee, Adam Roberts, Barret Zoph, Denny Zhou, Jason Wei, Kevin Robinson, David Mimno, and Daphne Ippolito. 2024. A pretrainer's guide to training data: Measuring the effects of data age, domain coverage, quality, & toxicity. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3245–3276, Mexico City, Mexico. Association for Computational Linguistics.

YINGWEI MA, Yue Liu, Yue Yu, Yuanliang Zhang, Yu Jiang, Changjian Wang, and Shanshan Li. At which training stage does code data help llms reasoning? In *The Twelfth International Conference on Learning Representations*.

Nestor Maslej, Loredana Fattorini, Raymond Perrault, Yolanda Gil, Vanessa Parli, Njenga Kariuki, Emily Capstick, Anka Reuel, Erik Brynjolfsson, John Etchemendy, Katrina Ligett, Terah Lyons, James Manyika, Juan Carlos Niebles, Yoav Shoham, Russell Wald, Tobi Walsh, Armin Hamrah, Lapo Santarlasci, and 4 others. 2025. Artificial intelligence index report 2025. *Preprint*, arXiv:2504.07139.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *EMNLP*.

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.

Benjamin Minixhofer, Edoardo M. Ponti, and Ivan Vulić. 2025. Zero-shot tokenizer transfer. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA. Curran Associates Inc.

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.

Arijit Nag, Soumen Chakrabarti, Animesh Mukherjee, and Niloy Ganguly. 2025. Efficient continual pretraining of LLMs for low-resource languages. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 3: Industry Track)*, pages 304–317, Albuquerque, New Mexico. Association for Computational Linguistics.

Gabriel Nicholas and Aliya Bhatia. 2023. Lost in translation: Large language models in non-english content analysis. *Preprint*, arXiv:2306.07377.

Ayomide Odumakinde, Daniel D'souza, Pat Verga, Beyza Ermis, and Sara Hooker. 2024. Multilingual arbitrage: Optimizing data pools to accelerate multilingual progress. *Preprint*, arXiv:2408.14960.

Jessica Ojo, Odunayo Ogundepo, Akintunde Oladipo, Kelechi Ogueji, Jimmy Lin, Pontus Stenetorp, and David Ifeoluwa Adelani. 2025. Afrobench: How good are large language models on african languages? *Preprint*, arXiv:2311.07978.

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

Jupinder Parmar, Sanjev Satheesh, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. 2024. Reuse, don't retrain: A recipe for continued pretraining of language models. *Preprint*, arXiv:2407.07263.

Guilherme Penedo, Hynek Kydlíček, Vinko Sabolčec, Bettina Messmer, Negar Foroutan, Martin Jaggi, Leandro von Werra, and Thomas Wolf. 2024. Fineweb2: A sparkling update with 1000s of languages.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2021. UNKs everywhere: Adapting multilingual language models to new scripts. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10186–10203, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

José Pombal, Dongkeun Yoon, Patrick Fernandes, Ian Wu, Seungone Kim, Ricardo Rei, Graham Neubig,

and André FT Martins. 2025. M-prometheus: A suite of open multilingual llm judges. *arXiv preprint arXiv:2504.04953*.

Alec Radford and Karthik Narasimhan. 2018. Improving language understanding by generative pre-training.

Kartik Ravisankar, Hyojung Han, and Marine Carpuat. 2025. Can you map it to english? the role of cross-lingual alignment in multilingual performance of llms. *Preprint*, arXiv:2504.09378.

David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. 2019. Experience replay for continual learning. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Winogrande: An adversarial winograd schema challenge at scale. *Preprint*, arXiv:1907.10641.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan LeBras, and Yejin Choi. 2019. Socialiqa: Commonsense reasoning about social interactions. *Preprint*, arXiv:1904.09728.

Craig W Schmidt, Varshini Reddy, Haoran Zhang, Alec Alameddine, Omri Uzan, Yuval Pinter, and Chris Tanner. 2024. Tokenization is more than compression. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 678–702, Miami, Florida, USA. 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.

Shaurya Sharthak, Vinayak Pahalwan, Adithya Kamath, and Adarsh Shirawalmath. 2025. Achieving tokenizer flexibility in language models through heuristic adaptation and supertoken learning. *Preprint*, arXiv:2505.09738.

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

Shivalika Singh, Yiyang Nan, Alex Wang, Daniel D'Souza, Sayash Kapoor, Ahmet Üstün, Sanmi Koyejo, Yuntian Deng, Shayne Longpre, Noah A. Smith, Beyza Ermis, Marzieh Fadaee, and Sara Hooker. 2025. The leaderboard illusion. *Preprint*, arXiv:2504.20879.

Shivalika Singh, Freddie Vargus, Daniel Dsouza, Börje F. Karlsson, Abinaya Mahendiran, Wei-Yin Ko, Herumb Shandilya, Jay Patel, Deividas Mataciunas, Laura OMahony, Mike Zhang, Ramith Hettiarachchi, Joseph Wilson, Marina Machado, Luisa Souza Moura, Dominik Krzemiński, Hakimeh Fadaei, Irem Ergün, Ifeoma Okoh, and 14 others. 2024. Aya dataset: An open-access collection for multilingual instruction tuning. *Preprint*, arXiv:2402.06619.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. 2024. Roformer: Enhanced transformer with rotary position embedding. *Neurocomput.*, 568(C).

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.

Chaofan Tao, Qian Liu, Longxu Dou, Niklas Muennighoff, Zhongwei Wan, Ping Luo, Min Lin, and Ngai Wong. 2024. Scaling laws with vocabulary: Larger models deserve larger vocabularies. In *Advances in Neural Information Processing Systems*, volume 37, pages 114147–114179. Curran Associates, Inc.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy Lillicrap, Angeliki Lazaridou, and 1332 others. 2025. Gemini: A family of highly capable multimodal models. *Preprint*, arXiv:2312.11805.

NLLB Team, 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, and 20 others. 2022. No language left behind: Scaling human-centered machine translation. *Preprint*, arXiv:2207.04672.

Ahmet Üstün, Viraat Aryabumi, Zheng 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. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15894–15939, Bangkok, Thailand. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Menan Velayuthan and Kengatharaiyer Sarveswaran. 2025. Egalitarian language representation in language models: It all begins with tokenizers. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 5987–5996, Abu Dhabi, UAE. Association for Computational Linguistics.

Shumin Wang, Yuexiang Xie, Bolin Ding, Jinyang Gao, and Yanyong Zhang. 2025. Language adaptation of large language models: An empirical study on LLaMA2. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 7195–7208, Abu Dhabi, UAE. Association for Computational Linguistics.

Thomas Wang, Adam Roberts, Daniel Hesslow, Teven Le Scao, Hyung Won Chung, Iz Beltagy, Julien Launay, and Colin Raffel. 2022. What language model architecture and pretraining objective works best for zero-shot generalization? In *International Conference on Machine Learning*, pages 22964–22984. PMLR.

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.

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, Aline Villavicencio, and Nikolaos Aletras. 2024a. An empirical study on cross-lingual vocabulary adaptation for efficient language model inference. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 6760–6785, Miami, Florida, USA. Association for Computational Linguistics.

Atsuki Yamaguchi, Aline Villavicencio, and Nikolaos Aletras. 2024b. How can we effectively expand the vocabulary of llms with 0.01gb of target language text? *Preprint*, arXiv:2406.11477.

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.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

A Additional Tokenizer Details

For both cluster and universal tokenizer training data weighting, we use a fixed proportion of 30% for English due to much larger data volume and much higher diversity in available data. This also ensures a fair comparison between tokenizer weighting strategies.

We use the GPT-4o (`gpt4-o200k`) regex for pretokenization in all the tokenizers we trained.* Tokenizer training data is sampled from the pretraining data mixture based on the weights described in Section 2.3. We use the tokenizers † library to train all the BPE models. We set the `min_frequency` argument to 5 on the BPE trainer to control the minimum frequency to merge pairs and do not use normalizers. Finally, we sample 50GB of data to train all the tokenizers.

B Compression Ratio

In order to intrinsically evaluate tokenizer quality, we measure compression ratio as compared to the publicly-available multilingual tokenizer used in Command-A (Cohere et al., 2025). Compression measures how efficiently data is represented in terms of size (in bytes), and BPE optimizes for this condition (Gage, 1994). Compression ratio compares compression values between tokenizers, and since lower compression is desirable, a compression ratio below 1 indicates that a tokenizer has more favorable compression. Previous work shows that compression correlates well with model performance, especially for generative tasks (Goldman et al., 2024; Gallé, 2019), although lower compression is not a sufficient condition for a better tokenizer (Schmidt et al., 2024). However, long sequence lengths are one of the ways in which inequitable treatment of languages begins at the tokenizer (Velayuthan and Sarveswaran, 2025; Ahia et al., 2023), and is therefore an important measure to consider along with downstream evaluations.

B.1 Impact of tokenizer language weighting on compression ratio

As a baseline, we evaluate uniform language weighting and compare it with our tokenizer where we use both data distribution and language bucketing strategies in conjunction. Compression ratios

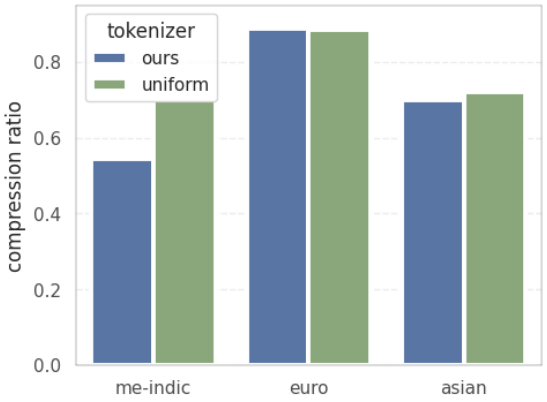


Figure 6: Compression ratios for our tokenizer and the baseline uniform tokenizer. Our tokenizer uses a special weighting that leverages training data distribution and the language grouping (§ 2.3), leading to a better compression (lower is better).

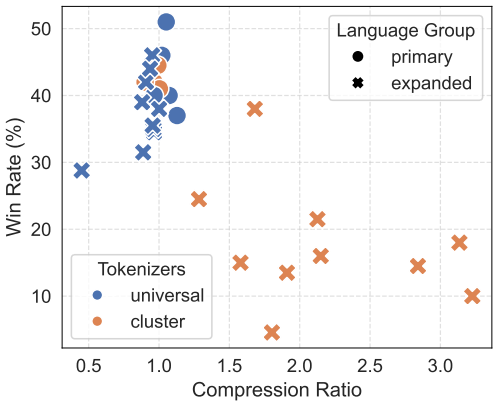


Figure 7: Adaptation results per language in Euro cluster together with tokenizers' compression ratio. While UNIVERSAL tokenizer enables better compression, especially for the expanded language subset, hence, better downstream performance, CLUSTER tokenizer fails to represent these languages, leading to lower adaptation results.

are computed against the multilingual tokenizer of open weight Command-A (Cohere et al., 2025) model on the test split of FineWeb-2 (Penedo et al., 2024).

Figure 6 shows the comparison. As seen, our tokenizer, which uses a special weighting, leads to better compression than the uniform baseline. Note that both of these tokenizers achieve overall higher compression performance than Command-A since they are trained with larger language coverage.

*https://github.com/openai/tiktoken/blob/4560a8896f5fb1d35c6f8fd6eee0399f9a1a27ca/tiktoken_ext/openai_public.py#L95

†https://github.com/huggingface/tokenizers

	bul	cat	ces	dan	deu	ell	est	eus	fin	fra	hrv	hun	ita	lit
UNIFORM	41.0	43.7	42.7	41.5	43.6	41.2	35.7	38.4	34.6	45.0	42.1	36.6	40.6	41.0
UNIVERSAL	42.1	46.5	45.5	41.1	44.3	42.7	37.9	40.6	32.4	45.3	42.6	37.9	40.1	43.1
	(+1.1)	(+2.8)	(+2.8)	(–0.4)	(+0.7)	(+1.5)	(+2.2)	(+2.2)	(–2.2)	(+0.3)	(+0.5)	(+1.3)	(–0.5)	(+2.1)

	lvs	nld	nob	pol	por	ron	rus	slk	slv	spa	srp	swe	ukr	Average
UNIFORM	42.3	42.7	42.3	38.4	41.0	40.8	41.0	42.4	39.5	42.5	42.4	42.9	40.9	41.0
UNIVERSAL	40.5	42.7	42.8	39.8	43.8	41.2	41.4	43.5	41.8	43.8	43.4	43.4	40.0	41.9
	(–1.8)	(+0.0)	(+0.5)	(+1.4)	(+2.8)	(+0.4)	(+0.4)	(+1.1)	(+2.3)	(+1.3)	(+1.0)	(+0.5)	(–0.9)	(+0.9)

Table 3: Comparison of UNIVERSAL vs. UNIFORM tokenizer performance on Belebele, when used for pretraining of the Euro cluster model.

B.2 Impact of compression ratio on downstream performance

Figure 7 explores the relationship between compression ratio and win rates for European cluster models trained with the UNIVERSAL and CLUSTER tokenizer on primary and expanded languages. The expansion languages exhibit large compression ratios with the CLUSTER, all over 1, which indicates that compression in that language is worse than the comparison tokenizer. At the same time, the win rates for those languages are lower than the primary languages, which also have a lower compression ratio. The UNIVERSAL tokenizer, however, shows relatively high win rates and low compression ratios for both primary and expansion languages. This result corroborates the relationship between compression ratio and downstream performance, and provides an additional dimension for language plasticity of the UNIVERSAL tokenizer.

C Additional Training Details

Our ablations are extensive and require a large amount of pretraining runs. Given the huge amount of compute required for pretraining, where training a 3.3B parameter model on 128 Nvidia H100 GPUs takes 11 hours, we focus on only 3.3 billion parameter language models for ablations. We train each base variant for 100 billion tokens using a total of 25,000 steps. Given the number of experiments we run, and the variety of factors we evaluate, this model size and amount of training steps is at the edge of what is computationally feasible at pretraining scale. Overall, the goal is not to emulate the settings of a full pretraining run, but to get sufficient signal about the relative merits of different approaches. In the continued pretraining strategy, we train for an additional 10.5B tokens and the targeted adaptation are done for 4 epochs over the respective datasets for each experiment.

Hyperparameters We performed a hyperparameter sweep on learning rate (LR), and used 2×10^{-2} as the peak LR for all pretraining experiments. We use a batch size of 512, a sequence length of 8192, and a cosine learning rate scheduler with a warmup of 2500 steps. For language adaptation experiments after pretraining, we use a constant LR of 1×10^{-4}, corresponding to the end LR of the pretraining stage.

D Additional Results

D.1 Additional pretraining results

Figure 8 shows pretraining results on the primary language subset (Euro cluster), measured in Belebele, throughout the pretraining run.

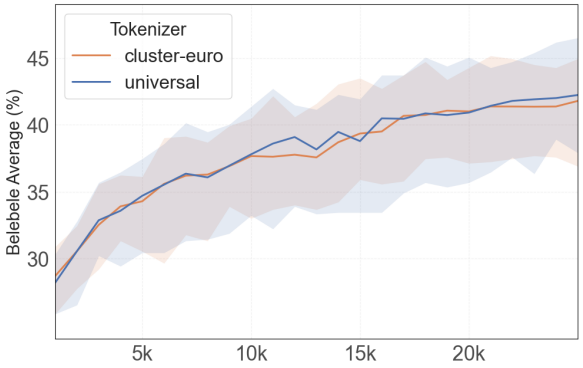


Figure 8: Average performance on primary languages during pretraining (Euro), measured in Belebele. Shaded areas indicate results across languages for each tokenizer. UNIVERSAL tokenizer shows nearly the same performance with CLUSTER tokenizer throughout the training.

		Asian			ME-Indic	
		CLUSTER	UNIVERSAL		CLUSTER	UNIVERSAL
PRIMARY	ind	56.5	46.5	arb	37.5	43.0
	jpn	52.0	40.5	heb	38.0	39.0
	kor	46.0	40.0	hin	41.5	38.0
	vie	49.5	39.5	pes	46.0	42.5
	zho	52.5	45.0	tur	35.5	38.5
EXPANDED	arb	16.0	27.0	deu	18.0	35.5
	deu	25.5	27.5	ell	10.6	29.5
	ell	9.0	29.0	fra	28.0	43.0
	fra	31.5	35.5	ind	24.5	46.5
	heb	6.1	24.5	jpn	20.5	46.0
	hin	5.1	27.5	kor	19.0	37.5
	pes	13.5	33.5	rus	20.5	41.0
	rus	17.5	38.0	spa	35.5	46.5
	spa	38.0	43.0	vie	16.0	38.0
	tur	13.0	27.5	zho	35.5	54.0

Table 4: Full win rate results for continued pretraining.

	Asian			ME-Indic	
	CLUSTER	UNIVERSAL		CLUSTER	UNIVERSAL
arb	16.5	30.5	deu	20.5	35.5
deu	20.0	30.5	ell	19.1	32.5
ell	13.1	32.0	fra	27.5	39.0
fra	27.5	37.0	ind	20.0	46.5
heb	12.1	32.0	jpn	16.5	41.5
hin	9.1	25.8	kor	16.5	37.0
pes	17.0	41.5	rus	22.5	39.0
rus	20.0	37.5	spa	39.5	46.5
spa	33.0	43.5	vie	16.5	43.0
tur	15.5	30.0	zho	34.5	50.5

Table 5: Targeted language adaptation win rates.

D.1.1 Continued pretraining

Table 4 presents the win rates by language after continued pretraining, divided into PRIMARY and EXPANDED languages.

D.1.2 Targeted adaptation

Table 5 presents the win rates by language after targeted adaptation, divided into PRIMARY and EXPANDED languages.

D.1.3 Higher-resourced unseen language

In order to see whether the benefits of a universal tokenizer hold in the case of unseen adaptation languages that are higher-resourced, we augment the Nepali language with 21,000 samples from Sangraha, an instruction-style dataset for Indic languages (Khan et al., 2024). We replicate the unseen language SFT experiment, post-training both the European cluster models trained with the UNIVERSAL tokenizer and the CLUSTER tokenizer. We find that in the case of greater data availability, the advantage of using a universal tokenizer increases, as seen in Table 6. On the smaller dataset, the UNIVERSAL tokenizer scores 3 points higher than the CLUSTER tokenizer. On the expanded data, the difference is 7%, with the UNIVERSAL showing win rate of 15.5%.

Category	Nepali Win Rate
Cluster	9%
Universal	12% (+3)
Cluster (extended data)	8.5%
Universal (extended data)	15.5% (+7)

Table 6: Nepali unseen language adaptation experiment, comparing different data-availability settings.

D.2 Judge Prompt for Win Rates

<system_prompt>
You are a helpful following assistant whose goal is to select the preferred (least wrong) output for a given instruction.

<user_prompt>
Which of the following answers is the best one for the given instruction? A good answer should follow these rules:
1) It should have correct reasoning,"
2) It should answer the request in the instruction,
3) It should be factually correct and semantically comprehensible,
4) It should be grammatically correct and fluent.

Instruction: instruction
Answer (A): completion_a
Answer (B): completion_b

FIRST provide a concise comparison of the two answers. If one answer is better, explain which you prefer and why. If both answers are identical or equally good or bad, explain why. SECOND, on a new line, state exactly one of 'Answer (A)' or 'Answer (B)' or 'TIE' to indicate your choice of preferred response.
Your response should use the format: Comparison: <concise comparison and explanation> Preferred: <'Answer (A)' or 'Answer (B)' or 'TIE'>.

E Languages

Table 7: Pretraining languages, including pretraining cluster assignment. Languages with a checkmark in the post-training column but without cluster assignment are used as unseen adaptation languages.

ISO Code	Language	Script	Family	Subgrouping	Resources	Cluster	In Post-Training
afr	Afrikaans	Latin	Indo-European	Germanic	Mid	-	✓
ara	Arabic	Arabic	Afro-Asiatic	Semitic	High	Middle East & Indic	✓
amh	Amharic	Ge'ez	Afro-Asiatic	Semitic	Low	-	✗
bel	Belarusian	Cyrillic	Indo-European	Balto-Slavic	Mid	-	✓
ben	Bengali	Bengali	Indo-European	Indo-Aryan	Mid	Middle East & Indic	✗
bul	Bulgarian	Cyrillic	Indo-European	Balto-Slavic	Mid	Euro	✗
cat	Catalan	Latin	Indo-European	Italic	High	Euro	✗
ces	Czech	Latin	Indo-European	Balto-Slavic	High	Euro	✓
cym	Welsh	Latin	Indo-European	Celtic	Low	Euro	✗
dan	Danish	Latin	Indo-European	Germanic	Mid	Euro	✗
deu	German	Latin	Indo-European	Germanic	High	Euro	✓
ell	Greek	Greek	Indo-European	Graeco-Phrygian	Mid	Euro	✓
eng	English	Latin	Indo-European	Germanic	High	Euro	✓
est	Estonian	Latin	Uralic	Finnic	Mid	Euro	✗
eus	Basque	Latin	Basque	-	High	Euro	✗
fil	Filipino	Latin	Austronesian	Malayo-Polynesian	Mid	Asian	✗
fin	Finnish	Latin	Uralic	Finnic	Mid	Euro	✗
fra	French	Latin	Indo-European	Italic	High	Euro	✓
gla	Scottish Gaelic	Latin	Indo-European	Celtic	Low	Euro	✗
gle	Irish	Latin	Indo-European	Celtic	Low	Euro	✗
glg	Galician	Latin	Indo-European	Italic	Mid	Euro	✗
guj	Gujarati	Gujarati	Indo-European	Indo-Aryan	Low	Middle East & Indic	✗
heb	Hebrew	Hebrew	Afro-Asiatic	Semitic	Mid	Middle East & Indic	✓
hin	Hindi	Devanagari	Indo-European	Indo-Aryan	High	Middle East & Indic	✓
hrv	Croatian	Latin	Indo-European	Balto-Slavic	High	Euro	✗
hun	Hungarian	Latin	Uralic	-	High	Euro	✗
hye	Armenian	Armenian	Indo-European	Armenic	Low	-	✓
ibo	Igbo	Latin	Atlantic-Congo	Benue-Congo	Low	-	✗
ind	Indonesian	Latin	Austronesian	Malayo-Polynesian	Mid	Asian	✓
ita	Italian	Latin	Indo-European	Italic	High	Euro	✓
jav	Javanese	Latin	Austronesian	Malayo-Polynesian	Low	Asian	✗
jpn	Japanese	Japanese	Japonic	Japanesic	High	Asian	✓
kaz	Kazakh	Cyrillic	Turkic	Common Turkic	Mid	-	✓
khm	Khmer	Khmer	Austroasiatic	Khmeric	Low	Asian	✗
kor	Korean	Hangul	Koreanic	Korean	Mid	Asian	✓
lao	Lao	Lao	Tai-Kadai	Kam-Tai	Low	Asian	✗
lav	Latvian	Latin	Indo-European	Balto-Slavic	Mid	Euro	✗
lit	Lithuanian	Latin	Indo-European	Balto-Slavic	Mid	Euro	✗
mlt	Maltese	Latin	Afro-Asiatic	Semitic	Low	Middle East & Indic	✗
msa	Malay	Latin	Austronesian	Malayo-Polynesian	Mid	Asian	✗
mya	Burmese	Myanmar	Sino-Tibetan	Burmo-Qiangic	Low	Asian	✗
nep	Nepali	Devanagari	Indo-European	Indo-Aryan	Low	-	✓
nld	Dutch	Latin	Indo-European	Germanic	High	Euro	✓
nor	Norwegian	Latin	Indo-European	Germanic	Low	Euro	✗
pan	Punjabi	Gurmukhi	Indo-European	Indo-Aryan	Low	Middle East & Indic	✗
pes	Persian	Arabic	Indo-European	Iranian	High	Middle East & Indic	✓
pol	Polish	Latin	Indo-European	Balto-Slavic	High	Euro	✓
por	Portuguese	Latin	Indo-European	Italic	High	Euro	✓
ron	Romanian	Latin	Indo-European	Italic	Mid	Euro	✓
rus	Russian	Cyrillic	Indo-European	Balto-Slavic	High	Euro	✓
sin	Sinhala	Sinhala	Indo-European	Indo-Aryan	Low	-	✓
slk	Slovak	Latin	Indo-European	Balto-Slavic	Mid	Euro	✗
slv	Slovenian	Latin	Indo-European	Balto-Slavic	Mid	Euro	✗
spa	Spanish	Latin	Indo-European	Italic	High	Euro	✓
srp	Serbian	Cyrillic	Indo-European	Balto-Slavic	High	Euro	✗
swa	Swahili	Latin	Atlantic-Congo	Benue-Congo	Low	-	✗
swe	Swedish	Latin	Indo-European	Germanic	High	Euro	✗
tam	Tamil	Tamil	Dravidian	South Dravidian	Mid	Middle East & Indic	✗
tel	Telugu	Telugu	Dravidian	South Dravidian	Low	Middle East & Indic	✗
tha	Thai	Thai	Tai-Kadai	Kam-Tai	Mid	Asian	✗
tur	Turkish	Latin	Turkic	Common Turkic	High	Middle East & Indic	✓
ukr	Ukrainian	Cyrillic	Indo-European	Balto-Slavic	Mid	Euro	✓
urd	Urdu	Arabic	Indo-European	Indo-Aryan	Mid	Middle East & Indic	✗
vie	Vietnamese	Latin	Austroasiatic	Vietic	High	Asian	✓
xho	Xhosa	Latin	Atlantic-Congo	Benue-Congo	Low	-	✗
yor	Yorùbá	Latin	Atlantic-Congo	Benue-Congo	Low	-	✗
yue	Cantonese	Han	Sino-Tibetan	Sinitic	Low	-	✓
zho	Mandarin Chinese	Han	Sino-Tibetan	Sinitic	High	Asian	✓
zul	Zulu	Latin	Atlantic-Congo	Benue-Congo	Low	-	✗