BACK TO THE ROOTS? RETHINKING MULTILINGUAL LLMS THROUGH FOUNDATIONAL LEARNER VOCABULARY

Anonymous authors
Paper under double-blind review

ABSTRACT

Multilingual large language models increasingly serve users across languages, yet their tokenizers fragment the lexical core: the words learners encounter first and rely on most. We introduce a new perspective that connects LLM vocabulary to language learner vocabulary, operationalized through LEXCORE, a 105-language foundational learner-vocabulary lexicon covering approximately 86.2% of the world's non-English-speaking population. Using LEXCORE, we study multilingual LLMs from diagnosis to intervention. Across 53 languages and 10 tokenizers, we show that contemporary tokenizers fragment most foundational learner vocabulary, with systematic disparities across resource levels, scripts, and language families. We further show that this fragmentation has a representation-level association: in bilingual retrieval, lexical grounding degrades as target words require more subword pieces, especially for lower-resource languages, while SUM emerges as the most stable composition operator. Building on these findings, we introduce LIFT, a learner-grounded, tokenizer-native vocabulary-injection method that reuses existing tokenizers, selects fragmented learner words under a fairness-aware maximin budget, and provides a worst-language relative-burden guarantee. LIFT improves several fairness-sensitive tokenization, fragmentation, and adoption metrics on FLORES+ and FineWeb2 without retraining tokenizers from scratch. Finally, in a controlled Gemma-3 4B continued-pretraining study, sum-initialized LIFT-3K achieves the best observed quality–parity–efficiency tradeoff among the controlled variants across multilingual QA, bidirectional translation, Information Parity, and decoding efficiency. These results show that foundational learner vocabulary is both a compact diagnostic lens for multilingual tokenizer inequity and a practical basis for targeted, budget-controlled adaptation of existing multilingual LLMs.

1 INTRODUCTION

Multilingual large language models (LLMs) now support an impressive range of languages and tasks, yet their multilingual competence remains uneven: performance, reliability, and language coverage are typically strongest for high-resource languages, especially English, and weaker for many others (Grattafiori et al., 2024; Üstün et al., 2024; Qin et al., 2025). Existing remedies often address this imbalance through multilingual continued pretraining (Penedo et al., 2025) or instruction tuning (Singh et al., 2024), after the tokenizer has already fixed the basic units of representation. We instead argue for rethinking multilingual LLMs from an earlier point in the pipeline: the lexical units they are asked to represent. Our lens is *foundational learner vocabulary*: the words learners encounter early, rely on heavily, and use to build communicative competence. Language pedagogy treats such vocabulary as central to early comprehension and broader lexical development (Nation, 2001; Webb & Nation, 2017; Schmitt, 2019). For multilingual LLMs, this vocabulary provides a compact but revealing lexical core: if the words learners need first are fragmented or weakly supported, multilingual competence is fragile at one of its most basic levels.

This perspective connects human language learning to multilingual language modeling. In second-language acquisition, learners relate a new language (*L2*) to a stronger first language (*L1*), especially when mapping early word forms to meaning (Odlin, 1989; Kroll & Stewart, 1994). We use this as an analogy, not as a claim that LLMs learn like humans: English functions as the lexical reference

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

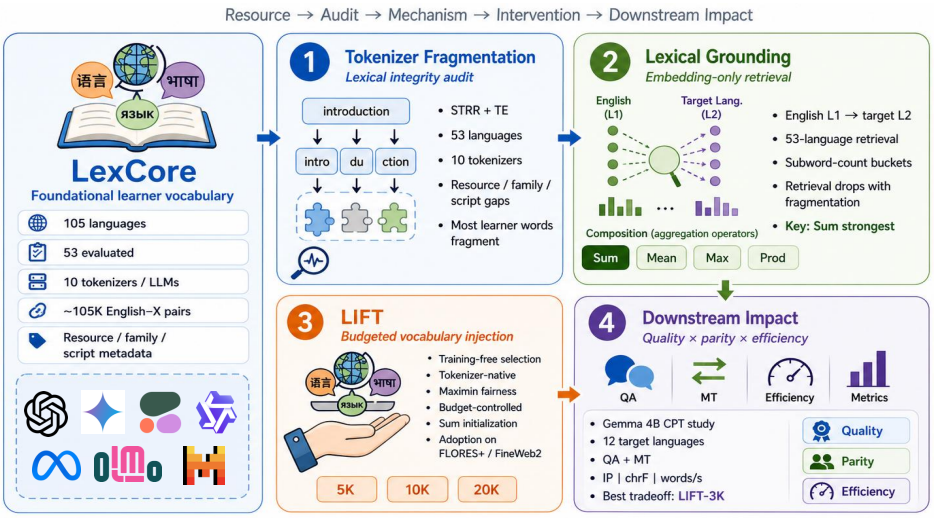


Figure 1: Overview of the paper. LEXCORE provides a 105-language foundational learner-vocabulary lexicon, with 53 languages evaluated in the main study. The four numbered panels summarize our proposed pipeline: **(1)** tokenizer fragmentation analysis, **(2)** lexical grounding via embedding-only retrieval, **(3)** LIFT as a budgeted learner-grounded vocabulary intervention, and **(4)** downstream evaluation of quality, parity, and efficiency.

point in multilingual modeling (Zhao et al., 2024), while target-language words must be represented through the model's shared tokenizer vocabulary. Unlike human learners, LLMs do not operate over words directly; their basic units are tokenizer entries (Mielke et al., 2021). When a foundational target-language word is split into multiple subword pieces while its English counterpart remains compact, the model must compose a fragmented lexical item before it can ground the word lexically or use it fluently. This motivates our central question: *Do multilingual LLMs support the words learners need first, and can this lexical core guide diagnosis, intervention, and downstream adaptation?*

Tokenization is the natural starting point for this question because it is not a neutral preprocessing step, but a core modeling decision that determines the discrete units over which LLMs operate (Mielke et al., 2021; Alqahtani et al., 2026). Modern subword tokenizers, including BPE and SentencePiece-style methods, are typically induced from corpus statistics (Gage, 1994; Sennrich et al., 2016; Kudo & Richardson, 2018); as a result, their vocabulary capacity tends to favor languages, scripts, and domains that dominate the training data. This can create systematic cross-lingual inequities: some languages require many more tokens to encode the same content (Ahia et al., 2023; Petrov et al., 2023), increasing fragmentation, latency, inference cost, and context-window pressure, with downstream consequences for multilingual quality and scalability (Lotz et al., 2025; Arnett et al., 2025). Yet practical remedies remain difficult. Training a new multilingual tokenizer can require model retraining or costly adaptation (Foroutan et al., 2026), while unconstrained vocabulary expansion scales poorly as the number of languages grows. This tension motivates a more targeted question: can we use a compact learner-vocabulary core to diagnose where existing tokenizers fail, expand them under a controlled budget, and improve multilingual behavior without rebuilding tokenizers from scratch?

We answer this question through the four-part pipeline in Figure 1. We introduce a new perspective that explicitly connects LLM vocabulary to language learner vocabulary, operationalized through LEXCORE: a 105-language foundational learner-vocabulary lexicon covering approximately 86.2% of the world's non-English-speaking population. Using a 53-language evaluation subset of LEXCORE, (1) we audit type-level lexical integrity across 10 tokenizers with STRR and TE, showing that even the strongest tokenizer preserves only about one-fifth of foundational learner vocabulary as atomic units, with disparities across resource levels, scripts, and language families. (2) We show that fragmentation has a representation-level association: in bilingual retrieval, performance degrades as target words require more subword pieces, especially for lower-resource languages, while SUM emerges as the most stable composition operator. (3) We propose LIFT , a learner-grounded, tokenizer-native injection method that reuses existing tokenizers rather than training new ones from scratch. LIFT selects shared fragmented surface forms under a global budget using a fairness-aware maximin objective, provides a worst-language relative-burden guarantee, and improves several fairness-sensitive tokenization,

fragmentation, and injected-token adoption metrics on FLORES+ and FineWeb2. ④ We validate downstream impact in a controlled Gemma-3 4B continued-pretraining study with a 2B-token budget: injected tokens are initialized from summed subword embeddings, are exposed to learning signal because LEXCORE words are well represented in the sampled adaptation corpora, and LIFT-3K yields the best observed quality–parity–efficiency tradeoff across QA, bidirectional translation, Information Parity, and decoding efficiency. Collectively, these findings show that foundational learner vocabulary is both a diagnostic lens for multilingual tokenizer inequity and a budget-controlled basis for improving existing multilingual LLMs through targeted vocabulary expansion.

2 EVALUATION SETUP

We operationalize our learner-vocabulary lens through LEXCORE, a multilingual lexicon covering 105 languages and approximately 105,000 English–target-language bilingual word pairs. The lexicon covers approximately 86.2% of the world's non-English-speaking population.[1] Its primary source is `1000MostCommonWords`,[2] a language-learning platform that curates lists of 1,000 common words per language for learners, with language-specific entries and translations, contributed by language experts worldwide and broad learner use. We use LEXCORE not as a complete model of lexical competence, but as a compact, standardized, and pedagogically meaningful approximation of the lexical core that many learners encounter first, rely on frequently, and use to build communicative competence. Appendix A validates its coverage using independent vocabulary resources and corpus evidence.

Throughout the paper, we use the 53 LEXCORE languages that overlap with the Aya Dataset (Singh et al., 2024), a widely used multilingual instruction-tuning resource with broad language coverage, making it a practical basis for selecting a diverse and experimentally tractable set of languages spanning resource levels, language families, and scripts. We evaluate 10 contemporary tokenizer/model families over this 53-language subset. The only exception is the downstream continued-pretraining study, where we restrict evaluation to the 12-language intersection of Global-MMLU-Lite (Singh et al., 2025) and FLORES+ (NLLB Team et al., 2024). This follows best practices for multilingual evaluation by prioritizing human-annotated QA data over machine-translated test items (Arnett, 2025), while ensuring a shared language set for QA and bidirectional machine translation. Appendix Table 21 provides full language metadata, and Table 19 provides model/tokenizer details.

3 LEXICAL INTEGRITY IN MULTILINGUAL TOKENIZERS

We first ask how well contemporary tokenizers preserve foundational learner vocabulary as atomic lexical units. This question is tokenizer-native: before studying downstream behavior, we need a diagnostic that measures what the tokenizer itself does to words learners encounter early and use frequently. We therefore evaluate lexical integrity at the *type level*, asking whether each learner word remains a single token or is fragmented into multiple pieces. This complements corpus-level metrics such as fertility or tokens-per-byte, which summarize behavior over running text but can compress allocation differences into a narrow numeric band (Rust et al., 2021; Ali et al., 2024). In contrast, a type-level diagnostic exposes how much direct lexical coverage is missing and where it occurs, making it interpretable and actionable for fairness-oriented multilingual tokenizer design (§5).

Single-Token Retention Rate (STRR). Let T denote a tokenizer and W_ℓ a *type-based* learner wordlist for language ℓ, consisting of distinct word forms, each counted once regardless of corpus frequency. For a word type $w \in W_\ell$, let $|T(w)|$ denote the number of tokens produced by applying tokenizer T to w. We define the *Single-Token Retention Rate* (STRR) as

$$\mathrm{STRR}(T; W_\ell) = \frac{1}{|W_\ell|} \sum_{i=1}^{|W_\ell|} \mathbf{1}(|T(w_i)| = 1) \times 100. \tag{1}$$

where $\mathbf{1}(\cdot)$ is the indicator function. STRR measures the percentage of learner word *types* preserved as atomic units under tokenization. Higher STRR indicates stronger lexical integrity, reflecting a tokenizer's ability to represent complete word forms without unnecessary fragmentation. Because

[1] Population coverage is estimated using Ethnologue data, accessed March 2026.

[2] `https://1000mostcommonwords.com`; see also the platform description on the About page.

Table 1: Tokenizer-level STRR summaries across 53 languages. Columns report vocabulary size, average STRR, TE, and mean STRR by resource class, language family, and script. Gemma-3 achieves the highest STRR (19.62%) and TE (3.62), but all tokenizers still fragment most learner vocabulary, with weaker retention for low-resource and non-Latin languages. The near-identical GPT-4 and OLMo-3 profiles reflect shared tokenizer provenance rather than independent tokenizer behavior: OLMo-3 uses a `cl100k`-compatible vocabulary with 100% ID/rank agreement over shared GPT-4 token surfaces. Group definitions, abbreviations, and counts are provided in Appendix Table 20 and full per-language STRR is presented in Appendix Table 13.

Tokenizers	Vocab Size	Avg. STRR	TE	Resource Class			Language Family					Language Scripts			
				High [17]	Mid [13]	Low [23]	IndoE [24]	NigerC [6]	AustN [5]	Drav [4]	Others [14]	Latn [28]	Arab [5]	Cyrl [4]	Others [16]
openai/GPT-4	100,277	8.59	1.72	12.42	5.87	7.30	8.27	10.67	11.76	0.15	9.52	13.45	6.42	2.59	2.26
openai/GPT-5	200,019	15.71	2.96	21.37	14.18	12.39	16.28	16.80	17.23	1.70	17.73	19.65	19.87	10.94	8.71
openai/gpt-oss-120b	200,019	15.71	2.96	21.37	14.18	12.39	16.28	16.80	17.23	1.70	17.73	19.65	19.87	10.94	8.71
CohereLabs/aya-expanse-32b	255,029	16.94	3.13	26.83	13.77	11.42	17.74	14.56	20.28	0.18	20.19	21.65	20.70	13.19	8.47
mistralai/Ministral-3-14B	131,072	12.58	2.46	18.57	10.72	9.21	12.74	11.68	14.00	0.75	15.58	15.78	18.52	9.12	5.99
meta-llama/Llama-3.1-70B	128,256	11.22	2.20	15.85	10.01	8.47	10.81	11.50	13.17	0.15	14.26	14.25	14.57	7.77	5.71
Qwen/Qwen3-32B	151,669	11.90	2.30	17.84	10.17	8.50	10.87	10.75	13.09	0.50	17.01	14.10	19.10	8.03	6.78
google/gemma-3-27b	262,145	**19.62**	**3.62**	27.88	19.77	13.43	20.77	14.58	20.75	4.75	23.67	21.35	20.40	12.31	18.17
allenai/Olmo-3-1125-32B	100,278	8.59	1.72	12.41	5.87	7.30	8.27	10.67	11.76	0.15	9.52	13.45	6.42	2.59	2.26
deepseek-ai/DeepSeek-V3.2	128,815	11.45	2.24	15.94	10.94	8.42	10.58	11.17	13.45	0.33	15.53	14.23	13.42	7.43	6.98

STRR is computed over types rather than token frequencies, it isolates tokenizer behavior from corpus usage statistics and provides a controlled cross-lingual diagnostic of lexical coverage. We also report Tokenizer Efficiency (TE), a vocabulary-size-normalized STRR summary; its formal definition and interpretation are provided in Appendix Section B.

3.1 Results and Analysis

Foundational learner vocabulary remains heavily fragmented. Table 1 summarizes tokenizer-level STRR across 53 languages, with full per-language STRR (Table 13) and token-length distributions (Table 14) of Appendix. Even the strongest tokenizer preserves only a small fraction of foundational learner vocabulary as atomic units: Gemma-3 achieves the highest average STRR (19.62%), followed by Aya-Expanse (16.94%) and GPT-5/GPT-OSS (15.71%). Thus, most learner words are still fragmented, including words central to early language use and comprehension. Vocabulary size alone does not explain this pattern: Aya-Expanse is comparable in size to Gemma-3 but lower in STRR, while GPT-5/GPT-OSS use about 200K entries and still fall behind. Lexical integrity therefore depends less on raw vocabulary scale than on how capacity is allocated across languages.

Coverage is uneven across resources, families, and scripts. The grouped results reveal persistent allocation asymmetries. High-resource languages receive much stronger single-token coverage than low-resource languages: Gemma-3 reaches 27.88% STRR for high-resource languages but only 13.43% for low-resource languages, while Aya-Expanse drops from 26.83% to 11.42%. Script and family effects are sharper: many non-Latin scripts remain heavily fragmented, and Dravidian languages form the clearest failure case. GPT-4, Llama-3.1, and OLMo-3 are near 0.15% STRR for Dravidian languages, Aya-Expanse reaches only 0.18%, and even Gemma-3 rises only to 4.75%.

Main finding and implication. Modern tokenizers fragment most foundational learner vocabulary unevenly: resource-rich, Latin-script, and some family groups receive stronger lexical integrity, while low-resource, non-Latin, and especially Dravidian languages remain under-covered. STRR exposes this pre-inference allocation bias and motivates our next question: whether fragmentation weakens multilingual lexical grounding (§4).

4 Compositional Lexical Grounding in Multilingual Embeddings

Our goal is to evaluate whether tokenizer fragmentation weakens lexical grounding in the model's native embedding space. Foundational learner vocabulary is often fragmented by multilingual tokenizers, as shown in Section 3; we therefore ask whether word-level cross-lingual structure can be recovered from token embeddings alone, without contextual layers, attention, or learned cross-lingual mappings. This design isolates the effect of tokenizer-induced segmentation and subword composition on bilingual lexical alignment, while building on evidence that token embeddings encode substantial lexical and cross-lingual structure (Lee et al., 2025; Kulkarni et al., 2026). In language-learning terms, English provides the *L1* anchor, while each target language is treated as the *L2* whose lexical grounding is evaluated (e.g., using English *house* as the anchor to retrieve French *maison*).

Let $\mathcal{L}$ denote the set of 53 target languages (discussed in Section 2). For each language $x \in \mathcal{L}$, let $\mathcal{P}_x$ denote the bilingual learner lexicon between English and x, defined as

$$\mathcal{P}_x = \{(w_i^{\text{en}}, w_i^x)\}_{i=1}^{N_x},$$

where w_i^{en} is an English learner word, w_i^x is its aligned form in language x, and $N_x = |\mathcal{P}_x|$. For a model with vocabulary V and input embedding matrix $E \in \mathbb{R}^{|V| \times d}$, let $e(t) \in \mathbb{R}^d$ denote the embedding of token $t \in V$. Each surface form is tokenized as

$$T(w) = (t_1, \ldots, t_k), \qquad k = |T(w)|.$$

Filtering and subword composition. To avoid trivial retrieval shortcuts, we retain only non-empty single-word forms on both sides and remove pairs in which the target-language form exactly matches an English candidate in the same language-specific lexicon. For each retained word w, we construct a word-level vector from its token embeddings:

$$v_C(w) = C(e(t_1), \ldots, e(t_k)) \in \mathbb{R}^d,$$

where C is a fixed composition operator. We evaluate SUM, MEAN, MAX, and PROD (elaborated in Appendix Table 8). All operators reduce to the original token embedding when $k = 1$, so their differences arise only when words require subword composition.

Embedding-only bilingual retrieval. For each language x and composition operator C, we perform bilingual retrieval over the aligned lexicon $\mathcal{P}_x$. Given an English query word w_i^{en}, we use its composed vector $v_C(w_i^{\text{en}})$ to search over all composed target-language candidates $\{v_C(w_j^x)\}_{j=1}^{N_x}$, including the aligned target form w_i^x. Retrieval is based on cosine similarity:

$$s_{ij} = \cos\big(v_C(w_i^{\text{en}}), v_C(w_j^x)\big)\,.$$

The correct target is the aligned item $j = i$. We report Precision@1 and Mean Reciprocal Rank (MRR). To connect retrieval behavior to tokenizer fragmentation, we stratify target-language words by token-length buckets:

$$|T(w^x)| \in \{1, 2, 3\text{–}4, 5+\}.$$

Thus, STRR measures how often a tokenizer preserves whole word types, whereas these target-token buckets measure how retrieval changes as target-side compositional burden increases.

Results and analysis. Figure 2 shows that SUM is the strongest and most stable embedding-only composition operator across models. MEAN remains competitive, but SUM provides the most reliable way to aggregate existing subword embeddings into word-level representations, whereas PROD consistently underperforms. This result is important for the next stage of the paper: if a fragmented word is later injected as a new vocabulary item, the sum of its existing subword embeddings provides a natural, model-native initialization (§6).

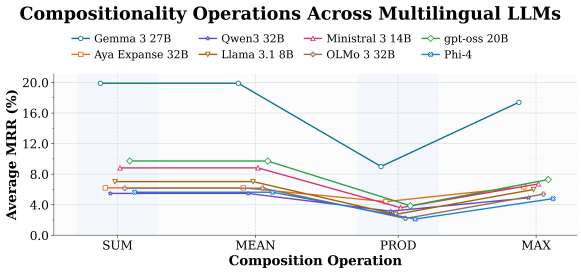


Figure 2: Composition-operation sensitivity across multilingual LLM embeddings, measured by average MRR.

Figure 3 links this compositional behavior to tokenizer fragmentation. Panel (a) shows that retrieval performance declines as the target lexical item requires more subword pieces: single-token targets are easiest to retrieve, while words split into three or more pieces show weaker grounding. Panel (b) shows that this degradation is resource-sensitive. High-resource languages retain stronger retrieval under mild fragmentation, whereas mid- and low-resource languages deteriorate more quickly as target-side token count increases. Together with the STRR analysis in Section 3, this establishes a representation-level link: tokenizer-level lexical preservation is associated with stronger performance on harder cross-lingual retrieval buckets. The P@1 results are shown in Appendix Figures 5 and 6.

Main finding and implication. SUM is the most stable composition operator, and retrieval degrades as target-side token length increases, especially under lower-resource conditions. Fragmented learner vocabulary is therefore harder to ground cross-lingually, motivating sum-based initialization for newly injected tokens.

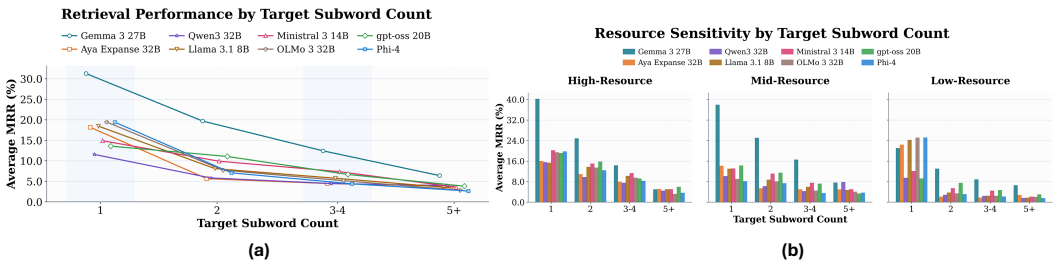


Figure 3: **Retrieval under target-side subword burden.** Average MRR decreases as target words require more subword pieces, both overall and by resource class, with sharper degradation in lower-resource languages. Corresponding P@1 results are shown in Figure 5 of Appendix.

5 LIFT: Budgeted Learner-Grounded Vocabulary Injection

Our objective is to turn the tokenizer diagnostic from the previous sections into a controlled intervention for reducing learner-vocabulary fragmentation. Given a global injection budget B, we ask which learner words should be added to an existing tokenizer so that fragmentation is reduced while benefits are distributed fairly across languages. We propose LIFT , a learner-grounded, tokenizer-native injection-selection method that is training-free at the selection stage. Rather than retraining a tokenizer from scratch, LIFT augments an existing tokenizer with selected learner-word tokens using only tokenizer-observable quantities: whether a word is fragmented, how severely it is fragmented, and which languages share the same surface form.

Design rationale. LIFT is designed around three constraints. First, it is diagnostic-grounded: only words identified as fragmented by STRR are eligible for injection. Second, it is budget-controlled: the global budget B explicitly limits vocabulary growth. Third, it is fairness-aware: because languages begin with different fragmentation burdens, LIFT equalizes relative burden reduction rather than raw token savings.

Candidate set and fragmentation burden. Since STRR identifies which learner words are not preserved as atomic units, it defines the injectable candidates for each language ℓ with learner vocabulary word list W_ℓ and tokenizer T:

$$\mathcal{C}_\ell = \{\, w \in W_\ell : |T(w)| > 1 \,\}, \qquad b_\ell(w) = |T(w)| - 1. \tag{2}$$

Here, $\mathcal{C}_\ell$ contains learner words that are not single-token under T, and $b_\ell(w)$ measures their excess segmentation beyond a single-token representation. The total unmet burden for language ℓ is

$$U_\ell = \sum_{w \in \mathcal{C}_\ell} b_\ell(w). \tag{3}$$

Thus, U_ℓ summarizes type-level excess segmentation over language ℓ's learner vocabulary.

Shared vocabulary and global budget. Because tokenizer vocabularies are shared across languages, LIFT injects unique surface forms rather than language-specific copies. Let $\mathcal{G}$ be the global set of candidate strings, and let $x_u \in \{0, 1\}$ indicate whether candidate $u \in \mathcal{G}$ is injected:

$$\mathcal{G} = \bigcup_{\ell \in \mathcal{L}} \mathcal{C}_\ell, \qquad \sum_{u \in \mathcal{G}} x_u \le B. \tag{4}$$

For each language ℓ, the burden associated with a global candidate u and the burden removed under a selection $\mathbf{x}$ are

$$b_\ell(u) = \begin{cases} |T(u)| - 1, & u \in \mathcal{C}_\ell, \\ 0, & \text{otherwise}, \end{cases} \qquad I_\ell(\mathbf{x}) = \sum_{u \in \mathcal{G}} x_u \, b_\ell(u). \tag{5}$$

Thus, LIFT is globally shared but language-aware: one injected string can benefit multiple languages when the same surface form appears across their candidate sets, while gains are tracked separately for each language.

Table 2: **Tokenizer-level comparison on FLORES+ and FineWeb2.** Lower is better for all metrics. LIFT improves several compression, fairness, and fragmentation metrics under controlled budgets; Greedy is an unconstrained large-vocabulary reference and excluded from best-value highlighting.

	FLORES+				FineWeb2			
Tokenizer	**CTC (K)↓**	**TP↓**	**TFG↓**	**MTP↓**	**TPB↓**	**SF↓**	**TFG↓**	**PCW↓**
Base tokenizers								
Gemma-3	**87.1**	0.656	0.112	1.402	0.229	2.283	0.170	0.661
GPT-OSS	87.5	0.683	0.107	2.170	**0.222**	2.310	0.135	0.688
Aya-Expanse	148.8	1.861	0.379	8.113	0.334	3.098	0.189	0.720
Qwen3	155.9	1.933	0.320	7.931	0.358	3.182	0.136	0.762
OLMo-3	180.7	2.460	0.350	10.637	0.414	3.734	0.134	0.805
Llama-3.1	158.5	2.033	0.358	10.641	0.351	3.266	0.190	0.759
Ministral-3	111.3	1.071	0.221	11.433	0.282	2.766	0.186	0.735
Our injected tokenizers								
Gemma-3 + Greedy (34,809)	92.3	0.603	0.086	1.170	0.243	2.085	0.154	0.553
Gemma-3 + LIFT-5,000	87.4	0.645	0.107	1.374	0.230	2.256	0.167	0.651
├── + LIFT-10,000	87.4	0.639	0.103	1.353	0.230	2.231	0.165	0.641
└── + LIFT-20,000	88.9	**0.626**	**0.094**	**1.289**	0.234	**2.176**	0.160	**0.611**
Aya-Expanse + Greedy (37,647)	138.6	1.351	0.269	5.941	0.339	2.725	0.091	0.591
Aya-Expanse + LIFT-5,000	147.0	1.804	0.372	7.966	0.332	3.051	0.184	0.709
├── + LIFT-10,000	144.6	1.742	0.364	7.789	0.329	3.002	0.179	0.699
└── + LIFT-20,000	140.2	1.604	0.342	7.271	0.324	2.897	0.166	0.670
Qwen3 + Greedy (38,846)	142.6	1.165	0.231	4.025	0.350	2.747	0.072	0.603
Qwen3 + LIFT-5,000	154.1	1.839	0.314	7.669	0.356	3.131	0.132	0.751
├── + LIFT-10,000	152.2	1.788	0.308	7.427	0.353	3.080	0.128	0.737
└── + LIFT-20,000	147.8	1.603	0.292	6.693	0.347	2.974	**0.120**	0.708

Relative burden reduction. Raw burden reduction is not comparable across languages because languages begin with different fragmentation levels. We therefore normalize by each language's initial unmet burden:

$$R_\ell(\mathbf{x}) = \frac{I_\ell(\mathbf{x})}{U_\ell}, \qquad \mathcal{L}^+ = \{\, \ell \in \mathcal{L} : U_\ell > 0 \,\}. \tag{6}$$

Without this normalization, languages with larger absolute fragmentation could dominate the budget simply because they offer larger raw gains.

Fairness-aware maximin injection. A utility-only objective would select candidates that remove the most total burden, but this can concentrate the budget on a small set of heavily fragmented languages and reinforce existing tokenizer inequities. LIFT instead maximizes the minimum relative burden reduction across languages using a maximin objective (Chen & Hooker, 2023):

$$\begin{aligned} \max_{\mathbf{x},\, t} \quad & t \\ \text{s.t.} \quad & \sum_{u \in \mathcal{G}} x_u \, b_\ell(u) \geq t \, U_\ell, \quad \forall \ell \in \mathcal{L}^+, \\ & \sum_{u \in \mathcal{G}} x_u \leq B, \\ & x_u \in \{0,1\}, \qquad \forall u \in \mathcal{G}, \\ & t \geq 0. \end{aligned} \tag{7}$$

Here, t is the minimum fraction of each language's initial burden guaranteed to be removed. Since $b_\ell(u)$ and U_ℓ are fixed for a given tokenizer, Eq. (7) is a mixed-integer linear program. The resulting injected vocabulary is budget-feasible, globally shared, learner-grounded, tokenizer-native, training-free in selection, and explicitly fairness-aware.

Guarantee and implementation. The maximin formulation gives an immediate worst-language guarantee. Let $t^\star$ denote the optimum of Eq. (7). Any optimal solution $\mathbf{x}^\star$ satisfies

$$R_\ell(\mathbf{x}^\star) \geq t^\star \qquad \forall \ell \in \mathcal{L}^+. \tag{8}$$

The proof follows directly from the fairness constraints in Eq. (7) and is provided in Appendix E. In practice, we compute STRR, construct each fragmented candidate set $\mathcal{C}_\ell$, form the global candidate set $\mathcal{G}$, solve Eq. (7), and inject the selected words up to budget B. Following the lexical-grounding result in Section 4, each injected token can be initialized as the sum of its existing subword embeddings. The full procedure is summarized in Appendix Algorithm 1.

5.1 Intrinsic Tokenization and Adoption Evaluation

Evaluation setup. We evaluate intrinsic tokenization efficiency in two complementary settings. On parallel crosslingual FLORES+ (NLLB Team et al., 2024), we report corpus token count (CTC), mean non-English token premium relative to English (TP), tokenizer fairness Gini (TFG), and maximum token premium (MTP). Following Arnett et al. (2025), we treat CTC on parallel text as the primary crosslingual compression metric and reserve byte-normalized measures for the non-parallel, natural-distribution FineWeb2 setting (Penedo et al., 2025). On FineWeb2, we report pooled tokens per byte (TPB), subword fertility (SF), tokenizer fairness Gini over per-language TPB, and proportion of continued words (PCW). As adoption evidence, we report injected token share (ITS) and injected type utilization (ITU), which measure whether injected tokens are actually used. Greedy variants add all fragmented candidates in this evaluation and are treated as an unconstrained large-vocabulary reference, while LIFT -k denotes our maximin MILP-based injection with budget k. Full definitions and implementation details are provided in Appendix C.

LIFT improves fragmentation under controlled budgets. Table 2 shows that LIFT improves intrinsic tokenization behavior without requiring the much larger Greedy expansions. For Gemma-3, LIFT-5K nearly preserves the base FLORES+ CTC (87.4K vs. 87.1K), while larger budgets improve fairness and fragmentation metrics such as TP, TFG, MTP, SF, and PCW. Aya-Expanse and Qwen3 show the same broad pattern: larger LIFT budgets steadily reduce FineWeb2 fragmentation metrics, although FLORES+ compression and fairness gains do not always improve monotonically. This indicates that vocabulary injection is not simply a matter of adding more tokens; budget size, candidate selection, and evaluation distribution jointly shape compression and fairness.

Tokenizer	FLORES+ ITS (%)↑	FLORES+ ITU (%)↑	FineWeb2 ITS (%)↑	FineWeb2 ITU (%)↑
Our injected tokenizers				
Gemma-3 + Greedy (34,809)	13.7	74.8	12.6	92.9
Gemma-3 + LIFT-5,000	1.3	61.3	1.2	91.4
|—— + LIFT-10,000	2.3	64.4	2.1	91.8
|—— + LIFT-20,000	**6.0**	70.0	**5.5**	**92.4**
Aya-Expanse + Greedy (37,647)	21.3	76.3	22.9	93.4
Aya-Expanse + LIFT-5,000	0.7	62.6	0.8	90.9
|——— + LIFT-10,000	1.5	65.6	1.5	91.5
|——— + LIFT-20,000	3.9	**70.3**	4.0	92.2
Qwen3 + Greedy (38,846)	22.8	76.9	24.2	93.6
Qwen3 + LIFT-5,000	0.9	64.7	0.9	91.2
|— + LIFT-10,000	1.8	66.5	1.9	91.8
|— + LIFT-20,000	4.0	70.2	4.0	**92.4**

Table 3: **Injected-vocabulary adoption on FLORES+ and FineWeb2.** Higher is better for all metrics. LIFT maintains high ITU under smaller budgets, Greedy is shown as an unconstrained large-vocabulary reference and excluded from best-value highlighting.

LIFT tokens are selective but actively used. Table 3 shows that injected tokens are used in both FLORES+ and FineWeb2. Greedy achieves higher ITS because it injects tens of thousands more tokens, but LIFT maintains strong ITU under smaller budgets: LIFT-20K reaches roughly 70% ITU on FLORES+ and above 92% ITU on FineWeb2 across Gemma-3, Aya-Expanse, and Qwen3. This suggests that LIFT does not add dormant vocabulary entries. Instead, it injects fewer learner-grounded tokens, a fraction of which are activated in evaluation corpora, providing a controlled alternative to unconstrained expansion (by resource results in Appendix Tables 17 and 18).

Main finding and implication. LIFT improves fragmentation and several fairness-sensitive tokenization metrics under smaller budgets, while maintaining high injected-type utilization. This supports learner-grounded, budgeted injection as a practical route to improving multilingual tokenization.

6 Downstream Impact of Learner-Grounded Injection

Our goal is to evaluate whether the tokenizer- and embedding-level findings yield measurable downstream improvements in multilingual behavior. We use Gemma-3 4B as the controlled testbed because Section 3 identifies Gemma as the strongest tokenizer family in STRR and TE, while still leaving substantial fragmentation to address. We conduct three full continued-pretraining runs: CPT with the original tokenizer, CPT + LIFT -3K, and CPT + LIFT -5K. For the LIFT variants, newly injected tokens are initialized as the *sum* of their subword embeddings, following Section 4, where SUM is identified as the most stable composition operator, rather than random initialization.

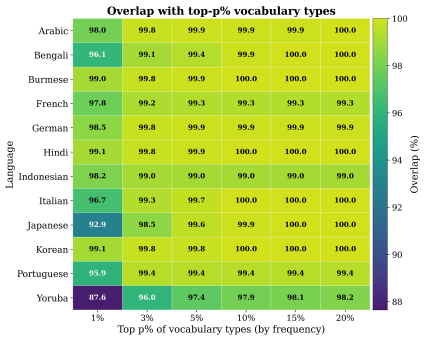


Figure 4: **Lexical overlap between LEXCORE and pretraining corpus.** For each of the 12 training-evaluation languages, the heatmap reports the fraction of lexicon entries covered by the top-p% most frequent vocabulary types in the pretraining corpus, with $p \in \{1, 3, 5, 10, 15, 20\}$. Most languages show high overlap even for small p, with near-saturation by the top 10–20% of vocabulary types. A complementary token-mass view is provided in Appendix Figure 9.

Table 4: **Downstream results for the Gemma-3 4B comparison with the baseline LLMs.** Higher is better for all metrics. CPT denotes continued pretraining; QA and IP are reported as percentages. Among Gemma-3 4B variants, CPT improves quality and parity over the original model, while LIFT yields further gains. LIFT -3K gives the strongest observed quality–parity–efficiency tradeoff under the 2B-token setting. Boldface marks the strongest Gemma-3 4B CPT variant in each column. Paired evaluation-level analyses are reported in Appendix F.2.

	Quality					Efficiency		
Model	**IP (%)↑**	**QA (%)↑**	**chrF↑ E→X**	**chrF↑ X→E**	**Avg.↑ chrF**	**tok/s↑ E→X**	**tok/s↑ X→E**	**Avg.↑ words/s**
Measured baselines								
Aya 8B	66.5	47.6	46.3	56.6	51.5	9.4	7.5	5.1
Gemma-3 4B	72.1	46.1	31.7	40.9	36.3	12.6	12.0	7.7
Llama 3.1 8B	65.7	46.6	18.5	51.7	35.1	12.7	12.4	2.8
Ministral 3B	67.4	55.9	5.8	41.6	23.7	6.5	11.3	4.9
Ministral 8B	69.9	61.7	26.1	33.4	29.8	8.6	9.3	6.4
Qwen3 4B	68.7	23.8	11.2	26.1	18.7	14.1	12.7	9.1
Qwen3 8B	69.8	25.0	11.5	26.6	19.1	11.2	10.4	7.0
Gemma-3 4B CPT variants								
Gemma-3 4B + CPT	74.6	48.1	35.2	43.4	39.3	**12.5**	**11.9**	7.80
+ LIFT-3K	**75.9**	**48.9**	**36.0**	**44.1**	**40.1**	12.2	11.6	**7.95**
+ LIFT-5K	75.4	48.6	35.7	43.6	39.7	11.9	11.3	7.75

To ensure consistent task coverage, we use Global-MMLU-Lite (Singh et al., 2025) for multilingual QA and FLORES+ (NLLB Team et al., 2024) for machine translation; their intersection yields 13 languages including English. We report QA accuracy, bidirectional chrF (Popović, 2015), Information Parity (IP) (Tsvetkov & Kipnis, 2024), and decoding efficiency. Each of the three continued-pretraining runs uses a fixed 2B-token budget with an 80:20 target-language-to-English replay mix, where English replay helps mitigate catastrophic forgetting (Elhady et al., 2025). All variants share the same sampled-text manifest, language mix, optimization recipe, and evaluation pipeline, isolating the contribution of vocabulary injection. Full training and evaluation details are provided in Appendix F.

Pretraining-data overlap. Before downstream evaluation, we verify that learner vocabulary is not simply absent from the adaptation corpora. Figure 4 shows lexical overlap between LEXCORE and high-frequency vocabulary types in the corresponding pretraining corpora. Most languages show high overlap even within the top few percent of vocabulary types, with near-saturation by the top 10–20%. Thus, the downstream challenge is not primarily lexical absence: high-coverage words can still be inefficiently segmented or weakly grounded, motivating tokenizer-aware adaptation.

Results and analysis. Table 4 shows that CPT improves the original Gemma-3 4B across parity and quality: IP rises from 72.1 to 74.6, QA from 46.1 to 48.1, and average chrF from 36.3 to 39.3, while preserving decoding efficiency. Adding LIFT further improves this tradeoff. LIFT -3K gives the strongest observed result, achieving the highest IP (75.9), QA (48.9), and average chrF (40.1), with gains in English→X and X→English translation. Paired evaluation-level tests show significant improvements over CPT for IP and translation, with a positive QA gain (Appendix F.2). LIFT-5K also improves over CPT on all quality metrics, but by smaller margins. This reveals a vocabulary-budget–adaptation-data tradeoff: expanding the vocabulary introduces more representations that must be sufficiently updated during continued pretraining, so larger injection budgets require more adaptation signal to be fully integrated (Mundra et al., 2024; Yamaguchi et al., 2026). Under the fixed 2B-token budget, LIFT-3K is better matched to the available data, while the 5K expansion likely requires additional continued pretraining data to realize its larger fragmentation reduction.

7 CONCLUSION

This paper examined multilingual LLMs through foundational learner vocabulary as a lens from diagnosis to intervention. Across tokenization, embedding-only retrieval, vocabulary injection, and downstream evaluation, learner vocabulary exposes systematic fragmentation, connects it to weaker lexical grounding, and guides intervention through LIFT. Under controlled budgets, LIFT improves several fairness-sensitive tokenization, fragmentation, and adoption metrics; in our Gemma-3 4B study, LIFT-3K achieves the best observed quality–parity–efficiency tradeoff. These findings position learner-grounded injection as a practical route for budget-controlled multilingual adaptation.

AI USE STATEMENT

We used generative AI tools in limited assistive roles. Specifically, they were used for **(1) text editing**, including improvements to grammar, spelling, phrasing, and word choice; **(2) code assistance**, including code fragments for data preprocessing and filtering, routine implementation tasks, and debugging; and **(3) visual assistance**, limited to generating decorative icons used in Figure 1.

All AI-assisted text, code, and visual materials were reviewed, adapted, and approved by the authors, and any generated or modified code was manually verified and tested before use. The authors developed the research questions and methodology, designed the experiments, conducted the analyses, interpreted the results, and determined the final presentation and conclusions. The authors take full responsibility for the final content of this work.

ETHICS STATEMENT

Data and research scope. This work studies multilingual tokenization, lexical representation, and vocabulary adaptation using publicly available research resources, including open models and tokenizers, publicly accessible lexical resources used to construct LEXCORE, pretraining corpora, and public evaluation benchmarks. The study does not involve human subjects, private communications, account-level metadata, or sensitive personal data. Lexical material collected from the web is limited to publicly accessible content required for research and excludes personal identifiers and login-protected data.

Linguistic scope and interpretation. LEXCORE uses foundational learner vocabulary as a controlled cross-lingual probe rather than a universal theory of lexical units. Word boundaries, morphology, dialectal variation, and community-specific usage differ across languages (Haspelmath, 2011), and single-token retention should therefore be interpreted as a diagnostic of tokenizer behavior rather than a ranking of languages or writing systems. English-anchored bilingual lexicons provide a common comparison framework but do not capture all multilingual lexical relations.

Intended use and release. LEXCORE is intended for tokenizer analysis, multilingual model diagnosis, vocabulary design, and controlled adaptation, not for high-stakes decisions concerning speakers, learners, educational placement, or language policy. We will release code, documentation, derived artifacts, and the processed LEXCORE resource under applicable upstream licensing terms, together with provenance and preprocessing details needed for reproducibility.

REPRODUCIBILITY STATEMENT

The paper and Appendix document LEXCORE construction and validation, tokenizer and model coverage, lexical-grounding experiments, the LIFT formulation and vocabulary-injection procedure, intrinsic and downstream evaluation, training configuration, prompts, metrics, statistical tests, and language-specific preprocessing. The three Gemma-3 4B continued-pretraining runs use the same sampled-text manifest, 2B-token budget, language mix, optimization recipe, and evaluation pipeline, enabling a controlled comparison of CPT, LIFT-3K, and LIFT-5K.

Experimental code and supporting artifacts are available at `anonymous.4open.science/r/LexCore-LIFT/`, including code for LEXCORE processing, tokenizer analysis, lexical grounding, LIFT optimization and vocabulary injection, evaluation, and analysis. A polished and fully documented release, together with the processed LEXCORE resource, will be made available upon paper's acceptance.

REFERENCES

Orevaoghene Ahia, Sachin Kumar, Hila Gonen, Jungo Kasai, David Mortensen, Noah Smith, and Yulia Tsvetkov. Do all languages cost the same? tokenization in the era of commercial language models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 9904–9923, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.614. URL https://aclanthology.org/2023.emnlp-main.614/.

Mehdi Ali, Michael Fromm, Klaudia Thellmann, Richard Rutmann, Max Lübbering, Johannes Leveling, Katrin Klug, Jan Ebert, Niclas Doll, Jasper Buschhoff, Charvi Jain, Alexander Weber, Lena Jurkschat, Hammam Abdelwahab, Chelsea John, Pedro Ortiz Suarez, Malte Ostendorff, Samuel Weinbach, Rafet Sifa, Stefan Kesselheim, and Nicolas Flores-Herr. Tokenizer choice for LLM training: Negligible or crucial? In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 3907–3924, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-naacl.247. URL https://aclanthology.org/2024.findings-naacl.247/.

Sawsan Alqahtani, Mir Tafseer Nayeem, Md Tahmid Rahman Laskar, Tasnim Mohiuddin, and M Saiful Bari. Stop taking tokenizers for granted: They are core design decisions in large language models. In Vera Demberg, Kentaro Inui, and Lluís Marquez (eds.), *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8410–8432, Rabat, Morocco, March 2026. Association for Computational Linguistics. ISBN 979-8-89176-380-7. doi: 10.18653/v1/2026.eacl-long.394. URL https://aclanthology.org/2026.eacl-long.394/.

Catherine Arnett. Best practices for open multilingual LLM evaluation. Hugging Face Blog, May 2025. URL https://huggingface.co/blog/catherinearnett/multilingual-best-practices. Community article. Accessed: 2026-05-05.

Catherine Arnett, Tyler A. Chang, Stella Biderman, and Ben Bergen. Explaining and mitigating crosslingual tokenizer inequities. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=0Y7AxxNCYh.

Charles Browne, Brent Culligan, and Joseph Phillips. The new general service list. https://www.newgeneralservicelist.com/new-general-service-list, 2013. Accessed 2026-05-22.

Violet Xinying Chen and J. N. Hooker. A guide to formulating fairness in an optimization model. *Annals of Operations Research*, pp. 1–39, 2023. ISSN 0254-5330. doi: 10.1007/s10479-023-05264-y. URL https://doi.org/10.1007/s10479-023-05264-y. Published online April 7, 2023.

Marco Cognetta, Tatsuya Hiraoka, Rico Sennrich, Yuval Pinter, and Naoaki Okazaki. An analysis of BPE vocabulary trimming in neural machine translation. In Shabnam Tafreshi, Arjun Akula, João Sedoc, Aleksandr Drozd, Anna Rogers, and Anna Rumshisky (eds.), *Proceedings of the Fifth Workshop on Insights from Negative Results in NLP*, pp. 48–50, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.insights-1.7. URL https://aclanthology.org/2024.insights-1.7/.

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, Acyr Locatelli, Hangyu Lin, Dwarak Talupuru, Bharat Venkitesh, David Cairuz, Bowen Yang, Tim Chung, Wei-Yin Ko, Sylvie Shang Shi, Amir Shukayev, Sammie Bae, Aleksandra Piktus, Roman Castagné, Felipe Cruz-Salinas, Eddie Kim, Lucas Crawhall-Stein, Adrien Morisot, Sudip Roy, Phil Blunsom, Ivan Zhang, Aidan Gomez, Nick Frosst, Marzieh Fadaee, Beyza Ermis, Ahmet Üstün, and Sara Hooker. Aya expanse: Combining research breakthroughs for a new multilingual frontier, 2024. URL https://arxiv.org/abs/2412.04261.

Ahmed Elhady, Eneko Agirre, and Mikel Artetxe. Emergent abilities of large language models under continued pre-training for language adaptation. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 32174–32186, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.1547. URL https://aclanthology.org/2025.acl-long.1547/.

Julen Etxaniz, Gorka Azkune, Aitor Soroa, Oier Lopez de Lacalle, and Mikel Artetxe. Do multilingual language models think better in English? In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pp. 550–564, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-short.46. URL https://aclanthology.org/2024.naacl-short.46/.

Negar Foroutan, Clara Meister, Debjit Paul, Joel Niklaus, Sina Ahmadi, Antoine Bosselut, and Rico Sennrich. Parity-aware byte-pair encoding: Improving cross-lingual fairness in tokenization. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics*, San Diego, California, USA, July 2026. Association for Computational Linguistics. URL https://arxiv.org/abs/2508.04796. To appear.

Philip Gage. A new algorithm for data compression. *C Users J.*, 12(2):23–38, February 1994. ISSN 0898-9788. URL https://dl.acm.org/doi/abs/10.5555/177910.177914.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, and Others. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Martin Haspelmath. The indeterminacy of word segmentation and the nature of morphology and syntax. *Folia Linguistica*, 45(1):31–80, 2011. doi: 10.1515/flin.2011.002. URL https://doi.org/10.1515/flin.2011.002.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. The state and fate of linguistic diversity and inclusion in the NLP world. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 6282–6293, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.560. URL https://aclanthology.org/2020.acl-main.560/.

Seungduk Kim, Seungtaek Choi, and Myeongho Jeong. Efficient and effective vocabulary expansion towards multilingual large language models, 2024. URL https://arxiv.org/abs/2402.14714.

J.F. Kroll and E. Stewart. Category interference in translation and picture naming: Evidence for asymmetric connections between bilingual memory representations. *Journal of Memory and Language*, 33(2):149–174, 1994. ISSN 0749-596X. doi: https://doi.org/10.1006/jmla.1994.1008. URL https://www.sciencedirect.com/science/article/pii/S0749596X84710084.

Taku Kudo and John Richardson. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In Eduardo Blanco and Wei Lu (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 66–71, Brussels, Belgium, November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-2012. URL https://aclanthology.org/D18-2012/.

Atharva Kulkarni, Jacob Mitchell Springer, Arjun Subramonian, and Swabha Swayamdipta. Disentangling geometry, performance, and training in language models, 2026. URL https://arxiv.org/abs/2602.20433.

Andrew Lee, Melanie Weber, Fernanda Viégas, and Martin Wattenberg. Shared global and local geometry of language model embeddings. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=aJDykpJAYF.

Bofang Li, Aleksandr Drozd, Tao Liu, and Xiaoyong Du. Subword-level composition functions for learning word embeddings. In Manaal Faruqui, Hinrich Schütze, Isabel Trancoso, Yulia Tsvetkov, and Yadollah Yaghoobzadeh (eds.), *Proceedings of the Second Workshop on Subword/Character LEvel Models*, pp. 38–48, New Orleans, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-1205. URL https://aclanthology.org/W18-1205/.

Jindřich Libovický and Jindřich Helcl. Lexically grounded subword segmentation. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 7403–7420, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.421. URL https://aclanthology.org/2024.emnlp-main.421/.

Tomasz Limisiewicz, Jiří Balhar, and David Mareček. Tokenization impacts multilingual language modeling: Assessing vocabulary allocation and overlap across languages. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 5661–5681, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.350. URL https://aclanthology.org/2023.findings-acl.350/.

Jonas F. Lotz, António V. Lopes, Stephan Peitz, Hendra Setiawan, and Leonardo Emili. Beyond text compression: Evaluating tokenizers across scales. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 32155–32173, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.1546. URL https://aclanthology.org/2025.acl-long.1546/.

Clara Meister. Tokeval: A tokenizer analysis suite, 2025. URL https://github.com/cimeister/tokenizer-analysis. GitHub repository, accessed April 19, 2026.

Sabrina J. Mielke, Zaid Alyafeai, Elizabeth Salesky, Colin Raffel, Manan Dey, Matthias Gallé, Arun Raja, Chenglei Si, Wilson Y. Lee, Benoît Sagot, and Samson Tan. Between words and characters: A brief history of open-vocabulary modeling and tokenization in nlp, 2021. URL https://arxiv.org/abs/2112.10508.

Nandini Mundra, Aditya Nanda Kishore Khandavally, Raj Dabre, Ratish Puduppully, Anoop Kunchukuttan, and Mitesh M Khapra. An empirical comparison of vocabulary expansion and initialization approaches for language models. In Libby Barak and Malihe Alikhani (eds.), *Proceedings of the 28th Conference on Computational Natural Language Learning*, pp. 84–104, Miami, FL, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.conll-1.8. URL https://aclanthology.org/2024.conll-1.8/.

I. S. P. Nation. *Learning Vocabulary in Another Language*. Cambridge Applied Linguistics. Cambridge University Press, Cambridge, 2001. URL https://www.cambridge.org/core/books/learning-vocabulary-in-another-language/491314AA1B451AD04F3536000F1C9F0D.

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, 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. Scaling neural machine translation to 200 languages. *Nature*, 630(8018):841–846, 2024. ISSN 1476-4687. doi: 10.1038/s41586-024-07335-x. URL https://doi.org/10.1038/s41586-024-07335-x.

Terence Odlin. *Language Transfer*. Cambridge Applied Linguistics. Cambridge University Press, Cambridge, 1989. ISBN 9781139524537. doi: 10.1017/CBO9781139524537. URL https://www.cambridge.org/core/books/language-transfer/0A19422B8005A10AFC282615FD542519.

Open Language Profiles. CEFR-J vocabulary profile, version 1.5. https://github.com/openlanguageprofiles/olp-en-cefrj, 2018. Accessed 2026-05-22.

Guilherme Penedo, Hynek Kydlíček, Vinko Sabolčec, Bettina Messmer, Negar Foroutan, Amir Hossein Kargaran, Colin Raffel, Martin Jaggi, Leandro Von Werra, and Thomas Wolf. Fineweb2: One pipeline to scale them all — adapting pre-training data processing to every language. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=jnRBe6zatP.

Qiwei Peng, Yekun Chai, and Anders Søgaard. Understanding subword compositionality of large language models. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 22524–22535, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.1146. URL https://aclanthology.org/2025.emnlp-main.1146/.

Aleksandar Petrov, Emanuele La Malfa, Philip Torr, and Adel Bibi. Language model tokenizers introduce unfairness between languages. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=78yDLKi95p.

Maja Popović. chrF: character n-gram F-score for automatic MT evaluation. In Ondřej Bojar, Rajan Chatterjee, Christian Federmann, Barry Haddow, Chris Hokamp, Matthias Huck, Varvara Logacheva, and Pavel Pecina (eds.), *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pp. 392–395, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/W15-3049. URL https://aclanthology.org/W15-3049/.

Matt Post. A call for clarity in reporting BLEU scores. In Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Christof Monz, Matteo Negri, Aurélie Névéol, Mariana Neves, Matt Post, Lucia Specia, Marco Turchi, and Karin Verspoor (eds.), *Proceedings of the Third Conference on Machine Translation: Research Papers*, pp. 186–191, Brussels, Belgium, October 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-6319. URL https://aclanthology.org/W18-6319/.

Libo Qin, Qiguang Chen, Yuhang Zhou, Zhi Chen, Yinghui Li, Lizi Liao, Min Li, Wanxiang Che, and Philip S. Yu. A survey of multilingual large language models. *Patterns*, 6(1):101118, 2025. ISSN 2666-3899. doi: https://doi.org/10.1016/j.patter.2024.101118. URL https://www.sciencedirect.com/science/article/pii/S2666389924002903.

Phillip Rust, Jonas Pfeiffer, Ivan Vulić, Sebastian Ruder, and Iryna Gurevych. How good is your tokenizer? on the monolingual performance of multilingual language models. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *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)*, pp. 3118–3135, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.243. URL https://aclanthology.org/2021.acl-long.243/.

Norbert Schmitt. Understanding vocabulary acquisition, instruction, and assessment: A research agenda. *Language Teaching*, 52(2):261–274, 2019. doi: 10.1017/S0261444819000053. URL https://www.cambridge.org/core/journals/language-teaching/article/understanding-vocabulary-acquisition-instruction-and-assessment-a-research-agenda/0EE8CF8625F140F79E708626D7D0DACB.

Lisa Schut, Yarin Gal, and Sebastian Farquhar. Do multilingual LLMs think in english? In *ICLR 2025 Workshop on Building Trust in Language Models and Applications*, 2025. URL https://openreview.net/forum?id=I8BOtOPcOv.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In Katrin Erk and Noah A. Smith (eds.), *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1715–1725, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1162. URL https://aclanthology.org/P16-1162/.

Shivalika Singh, Freddie Vargus, Daniel D'souza, Börje F. Karlsson, Abinaya Mahendiran, Wei-Yin Ko, Herumb Shandilya, Jay Patel, Deividas Mataciunas, Laura O'Mahony, Mike Zhang, Ramith Hettiarachchi, Joseph Wilson, Marina Machado, Luisa Moura, Dominik Krzemiński, Hakimeh Fadaei, Irem Ergun, Ifeoma Okoh, Aisha Alaagib, Oshan Mudannayake, Zaid Alyafeai, Vu Chien, Sebastian Ruder, Surya Guthikonda, Emad Alghamdi, Sebastian Gehrmann, Niklas Muennighoff, Max Bartolo, Julia Kreutzer, Ahmet Üstün, Marzieh Fadaee, and Sara Hooker. Aya dataset: An open-access collection for multilingual instruction tuning. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11521–11567, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.620. URL https://aclanthology.org/2024.acl-long.620/.

Shivalika Singh, Angelika Romanou, Clémentine Fourrier, David Ifeoluwa Adelani, Jian Gang Ngui, Daniel Vila-Suero, Peerat Limkonchotiwat, Kelly Marchisio, Wei Qi Leong, Yosephine Susanto, Raymond Ng, Shayne Longpre, Sebastian Ruder, Wei-Yin Ko, Antoine Bosselut, Alice

Oh, Andre Martins, Leshem Choshen, Daphne Ippolito, Enzo Ferrante, Marzieh Fadaee, Beyza Ermis, and Sara Hooker. Global MMLU: Understanding and addressing cultural and linguistic biases in multilingual evaluation. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 18761–18799, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.919. URL https://aclanthology.org/2025.acl-long.919/.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, Gaël Liu, Francesco Visin, Kathleen Kenealy, Lucas Beyer, Xiaohai Zhai, Anton Tsitsulin, Robert Busa-Fekete, Alex Feng, Noveen Sachdeva, Benjamin Coleman, Yi Gao, Basil Mustafa, Iain Barr, Emilio Parisotto, David Tian, Matan Eyal, Colin Cherry, Jan-Thorsten Peter, Danila Sinopalnikov, Surya Bhupatiraju, Rishabh Agarwal, Mehran Kazemi, Dan Malkin, Ravin Kumar, David Vilar, Idan Brusilovsky, Jiaming Luo, Andreas Steiner, Abe Friesen, Abhanshu Sharma, Abheesht Sharma, Adi Mayrav Gilady, and Others. Gemma 3 technical report, 2025. URL https://arxiv.org/abs/2503.19786.

Yukio Tono (ed.). *The CEFR-J Handbook: A Resource Book for Using CAN-DO Descriptors for English Language Teaching*. Taishukan Shoten, Tokyo, 2013. URL https://ci.nii.ac.jp/ncid/BB12652677.

Alexander Tsvetkov and Alon Kipnis. Information parity: Measuring and predicting the multilingual capabilities of language models. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 7971–7989, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.468. URL https://aclanthology.org/2024.findings-emnlp.468/.

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. Aya model: An instruction finetuned open-access multilingual language model. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15894–15939, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.845. URL https://aclanthology.org/2024.acl-long.845/.

S. Webb and P. Nation. *How Vocabulary is Learned*. Oxford handbooks for language teachers. Oxford University Press, 2017. ISBN 9780194403559. URL https://books.google.ca/books?id=OwkjvgAACAAJ.

Chris Wendler, Veniamin Veselovsky, Giovanni Monea, and Robert West. Do llamas work in English? on the latent language of multilingual transformers. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15366–15394, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.820. URL https://aclanthology.org/2024.acl-long.820/.

Atsuki Yamaguchi, Aline Villavicencio, and Nikolaos Aletras. How can we effectively expand the vocabulary of llms with 0.01gb of target language text? *Computational Linguistics*, 52(1):295–330, 03 2026. ISSN 0891-2017. doi: 10.1162/COLI.a.581. URL https://doi.org/10.1162/COLI.a.581.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger

Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Yiran Zhao, Wenxuan Zhang, Guizhen Chen, Kenji Kawaguchi, and Lidong Bing. How do large language models handle multilingualism? In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=ctXYOoAgRy.

Supplementary Material: Appendices

Appendix Contents

A Validation of LEXCORE

B Tokenizer Efficiency

C Intrinsic Tokenization Metrics

D Frequency-Controlled Analysis of Lexical Retrieval

E Proof of the LIFT Guarantee

F Downstream Experimental Setup

F.1 Compute and Reproducibility
F.2 Robustness of Downstream Results
F.3 Benchmark Details
F.4 Prompting and Decoding Details

G Language-Specific Word Segmentation

H Composition of the 53-Language Evaluation Set

I Supplementary Literature Review

J Implications for Multilingual Tokenizer Design

K Limitations

A VALIDATION OF LEXCORE

LEXCORE operationalizes our learner-vocabulary perspective through a multilingual lexical inventory designed to capture words that are foundational for early language learning. We evaluate this resource from three complementary perspectives: **(1)** source and expert provenance, **(2)** agreement with independent learner-oriented vocabulary resources, and **(3)** agreement with corpus usage.

Motivation and lexical grounding. Our use of foundational learner vocabulary follows the paper's central language-learning perspective. In second-language acquisition, learners often relate early words in a new language (*L2*) to lexical concepts already available through a stronger first language (*L1*). For example, an English speaker learning Japanese may first connect mizu to "water" or inu to "dog". We use this only as an analogy, not as a claim that LLMs learn like humans. In multilingual language models, prior work has similarly identified English-aligned internal representations and English-pivot behavior in multilingual processing (Zhao et al., 2024; Wendler et al., 2024). In our setting, English therefore serves as a shared lexical reference point, while each target language provides the corresponding learner-oriented lexical inventory.

This choice gives LEXCORE three useful properties. First, it is **human-centered and pedagogically motivated**, because the vocabulary is defined by learner relevance rather than by model-specific corpus statistics. Second, it is **cross-lingually aligned**, providing English–target lexical pairs across 105 languages. Third, it is **empirically grounded**, as shown by its agreement with independent learner-vocabulary resources and its strong representation in multilingual corpus data. We therefore use learner vocabulary as a compact, interpretable, and cross-lingually comparable lexical core that complements corpus-level frequency and compression measures.

Source and expert provenance. The primary source of LEXCORE is `1000MostCommonWords`, a language-learning resource that provides approximately 1,000 foundational words per language together with English translations. The platform reports contributions from more than 100 language

experts worldwide and broad use by language learners.[3] This provides a pedagogically motivated starting point for constructing a multilingual lexical probe that is independent of any particular tokenizer or pretraining corpus.

Table 5: Agreement between the English counterpart inventories in LEXCORE and two independent learner-oriented vocabulary resources across all 105 languages. Coverage is computed separately for each language-specific English inventory and summarized across languages.

Independent resource	Role	Mean coverage	Minimum coverage
CEFR-J	Proficiency-oriented learner vocabulary	94.52%	87.17%
NGSL	Core learner/ESL vocabulary	87.10%	78.00%
Macro average	**Combined validation**	**90.81%**	**82.59%**

Agreement with independent learner-vocabulary resources. Every target-language entry in LEXCORE is aligned with an English counterpart. This makes it possible to evaluate the learner-vocabulary content of the full 105-language inventory.

For each language, we compare its aligned English inventory against two independent learner-oriented English vocabulary resources: the New General Service List (NGSL) (Browne et al., 2013) and CEFR-J (Tono, 2013; Open Language Profiles, 2018). NGSL represents a core vocabulary for English-language learners, while CEFR-J provides a proficiency-oriented learner-vocabulary reference. Table 5 summarizes the coverage across all 105 languages.

Across the 105 languages, the combined validation score has a mean of **90.81%**, a median of **90.89%**, and a minimum of **82.59%**. The consistently high agreement with independently constructed learner-vocabulary resources indicates that the English counterpart inventories in LEXCORE capture vocabulary strongly associated with foundational language learning.

This analysis evaluates the learner-oriented character of the aligned English inventories. It does not independently assess the correctness of every target-language translation, which remains a separate dimension of lexicon quality.

Agreement with corpus usage. The learner-oriented validation is complemented by corpus-based evidence from the continued-pretraining data. As shown in Figures 4 and 9, LEXCORE words overlap strongly with high-frequency vocabulary types in the corresponding multilingual corpora and account for substantial running token mass. This provides an independent usage-based signal that the vocabulary captured by LEXCORE is not only pedagogically motivated, but also frequently encountered in naturally occurring text.

In summary, source provenance, independent learner-resource agreement, and corpus usage provide complementary evidence that LEXCORE captures a meaningful foundational lexical core for multilingual tokenizer analysis.

B TOKENIZER EFFICIENCY

Raw STRR (§3) must be interpreted in light of vocabulary scale: a tokenizer with a much larger vocabulary has more representational capacity, but that capacity may not be distributed equitably across languages. To summarize tokenizer behavior across a language set $\mathcal{L}$ while accounting for vocabulary size, we define the mean STRR and *Tokenizer Efficiency* (TE) as

$$\bar{s}(T) = \frac{1}{|\mathcal{L}|} \sum_{\ell \in \mathcal{L}} \mathrm{STRR}(T; W_\ell), \qquad \mathrm{TE}(T) = \frac{\bar{s}(T)}{\log_{10}\big(V(T)\big)}, \tag{9}$$

where $V(T)$ denotes the tokenizer vocabulary size. TE contextualizes type-level lexical retention by vocabulary scale. Rather than penalizing large vocabularies, it asks how efficiently a tokenizer uses

[3] https://1000mostcommonwords.com; see also the platform description on the About page. The platform states that it was founded in 2014, receives hundreds of thousands of learner visits each year, and developed word lists with contributions from language experts around the world, including over 100 language experts.

its available capacity to preserve whole-word units across languages. A tokenizer with high STRR and high TE is therefore both more lexically complete and more efficient in its vocabulary allocation.

C Intrinsic Tokenization Metrics

We evaluate intrinsic tokenization efficiency in two complementary settings: a parallel cross-lingual setting based on FLORES+ (NLLB Team et al., 2024)[4] and a natural-distribution setting based on FineWeb2 (Penedo et al., 2025). Let $T(\cdot)$ denote a tokenizer, let $|T(x)|$ be the number of tokens produced for text x, and let $b(x)$ denote the number of UTF-8 bytes in x.

FLORES+: Parallel Crosslingual Compression. For each language $\ell \in L$, let

$$\mathcal{D}_\ell = \{x_1^\ell, \ldots, x_N^\ell\}$$

denote the aligned FLORES+ corpus for language ℓ (NLLB Team et al., 2024). Because FLORES+ is parallel, token counts are directly comparable across languages. Following Arnett et al. (2025), CTC on parallel text is treated as the primary crosslingual compression metric, while byte-normalized quantities are retained only for the non-parallel FineWeb2 setting.

Corpus Token Count (CTC). We define the corpus token count of language ℓ as

$$\mathrm{CTC}_\ell = \sum_{x \in \mathcal{D}_\ell} |T(x)|.$$

This quantity captures the total number of tokenizer outputs required to encode content-matched text in language ℓ. We summarize tokenizer-level compression by the average language-level corpus token count

$$\mathrm{CTC} = \frac{1}{|L|} \sum_{\ell \in L} \mathrm{CTC}_\ell,$$

where lower values indicate greater compression efficiency on parallel text (Arnett et al., 2025).

Token Premium Relative to English (TP). Let ℓ_{en} denote English. We define the token premium of language ℓ relative to English as

$$\mathrm{TP}_\ell = \frac{\mathrm{CTC}_\ell}{\mathrm{CTC}_{\ell_{\mathrm{en}}}} - 1.$$

This quantity measures how much more expensive language ℓ is to encode than English on matched content. We report the mean non-English token premium

$$\mathrm{TP} = \frac{1}{|L| - 1} \sum_{\ell \in L \setminus \{\ell_{\mathrm{en}}\}} \mathrm{TP}_\ell,$$

so that lower values indicate smaller average disadvantage relative to English (Petrov et al., 2023; Ali et al., 2024).

Tokenizer Fairness Gini (TFG). To quantify inequality in tokenization cost across languages, we use the Gini coefficient over language-level costs. For non-negative values $c_1, \ldots, c_n$ with mean

$$\mu = \frac{1}{n} \sum_{i=1}^{n} c_i,$$

the Gini coefficient is

$$\mathrm{Gini}(c_1, \ldots, c_n) = \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} |c_i - c_j|}{2n^2 \mu}.$$

For FLORES+, we define

$$\mathrm{TFG} = \mathrm{Gini}(\mathrm{CTC}_1, \ldots, \mathrm{CTC}_{|L|}),$$

where lower values indicate more even tokenization cost across languages (Meister, 2025)[5].

[4] https://huggingface.co/datasets/openlanguagedata/flores_plus

[5] https://github.com/cimeister/tokenizer-analysis-suite

Maximum Token Premium (MTP). To characterize worst-case disparity relative to English, we define

$$\mathrm{MTP} = \max_{\ell \in L \setminus \{\ell_{\mathrm{en}}\}} \mathrm{TP}_\ell.$$

Lower values indicate a smaller worst-language penalty (Petrov et al., 2023; Ali et al., 2024).

FineWeb2: Natural-Distribution Tokenization Efficiency. For each language $\ell \in L_{\mathrm{FW}}$, let $\mathcal{C}_\ell$ denote the corresponding FineWeb2 corpus (Penedo et al., 2025). Since FineWeb2 is not parallel, we do not interpret raw token counts as a controlled crosslingual compression signal. Instead, we report corpus-level efficiency and fragmentation statistics on natural-distribution text.

Tokens per Byte (TPB). We define the byte-normalized token cost of language ℓ as

$$\mathrm{TPB}_\ell = \frac{\sum_{x \in \mathcal{C}_\ell} |T(x)|}{\sum_{x \in \mathcal{C}_\ell} b(x)}.$$

This quantity measures corpus-level encoding efficiency normalized by raw text size. We report the pooled value

$$\mathrm{TPB} = \frac{\sum_{\ell \in L_{\mathrm{FW}}} \sum_{x \in \mathcal{C}_\ell} |T(x)|}{\sum_{\ell \in L_{\mathrm{FW}}} \sum_{x \in \mathcal{C}_\ell} b(x)},$$

where lower values indicate more efficient byte-normalized encoding.

Subword Fertility (SF). Let W_ℓ denote the multiset of segmented words in language ℓ. We define

$$\mathrm{SF}_\ell = \frac{\sum_{w \in W_\ell} |T(w)|}{|W_\ell|},$$

and report the pooled value

$$\mathrm{SF} = \frac{\sum_{\ell \in L_{\mathrm{FW}}} \sum_{w \in W_\ell} |T(w)|}{\sum_{\ell \in L_{\mathrm{FW}}} |W_\ell|}.$$

Subword fertility is the average number of tokenizer pieces per word; lower values indicate less over-segmentation (Rust et al., 2021).

Proportion of Continued Words (PCW). We define

$$\mathrm{PCW}_\ell = \frac{\sum_{w \in W_\ell} \mathbf{1}[|T(w)| > 1]}{|W_\ell|},$$

and report the pooled value

$$\mathrm{PCW} = \frac{\sum_{\ell \in L_{\mathrm{FW}}} \sum_{w \in W_\ell} \mathbf{1}[|T(w)| > 1]}{\sum_{\ell \in L_{\mathrm{FW}}} |W_\ell|}.$$

This measures the fraction of words split into multiple pieces; lower values indicate more whole-word coverage (Rust et al., 2021).

Tokenizer Fairness Gini on FineWeb2 (TFG). To quantify crosslingual inequality under natural-distribution text, we compute the Gini coefficient over per-language TPB values:

$$\mathrm{TFG}_{\mathrm{FW}} = \mathrm{Gini}(\mathrm{TPB}_1, \ldots, \mathrm{TPB}_{|L_{\mathrm{FW}}|}).$$

Lower values indicate more uniform byte-normalized tokenization cost across languages (Meister, 2025).

Supporting Adoption Metrics. As a supporting analysis, we measure whether injected vocabulary entries are actively used in practice. Let V_{inj} denote the set of injected vocabulary types.

Injected Token Share (ITS). For a corpus $\mathcal{C}$, injected token share is

$$\mathrm{ITS}(\mathcal{C}) = \frac{\#\{\text{token instances in } \mathcal{C} \text{ whose ids are in } V_{\mathrm{inj}}\}}{\#\{\text{all token instances in } \mathcal{C}\}}.$$

This quantifies how much of the produced token mass comes from injected entries.

Table 6: Frequency-controlled regression of lexical retrieval quality on target token count. Negative coefficients indicate lower retrieval quality as fragmentation increases, after controlling for FineWeb2 word frequency, model identity, and language identity.

Outcome	Token-count coefficient	Language-clustered SE	p-value
Reciprocal rank	-0.0133	0.0029	2.42×10^{-5}
P@1	-0.0109	0.0027	1.50×10^{-4}

Table 7: Retrieval performance by within-language frequency tertile and target-side fragmentation. In every frequency group, retrieval is substantially lower for words requiring five or more tokens than for single-token words.

Frequency tertile	MRR: 1 token	MRR: 5+ tokens	P@1: 1 token	P@1: 5+ tokens
Low	0.4325	0.0211	0.3855	0.0060
Mid	0.2759	0.0190	0.2343	0.0054
High	0.2016	0.0157	0.1563	0.0043

Injected Type Utilization (ITU). We define injected type utilization as

$$\mathrm{ITU}(\mathcal{C}) = \frac{\#\{\text{distinct injected token types observed in } \mathcal{C}\}}{|V_{\mathrm{inj}}|}.$$

This measures how much of the injected vocabulary is activated at least once. ITS and ITU are not direct quality metrics; rather, they provide supporting evidence that the injected vocabulary is meaningfully used in the evaluation corpora (Foroutan et al., 2026).

D Frequency-Controlled Analysis of Lexical Retrieval

The retrieval analysis in Section 4 establishes an association between target-side fragmentation and weaker bilingual lexical retrieval. Because more fragmented words may also differ systematically in corpus frequency, we further test whether word frequency explains this pattern.

Frequency-controlled regression. We analyze 877,176 word–model–language retrieval instances. For each target word, we obtain its corpus frequency from FineWeb2 and predict retrieval quality from the target token count while controlling for log word frequency, model identity, and language identity. Standard errors are clustered by language.

The token-count coefficient remains negative and statistically significant for both reciprocal rank and P@1 after controlling for corpus frequency. Replacing raw frequency with each word's within-language frequency percentile yields nearly identical estimates: -0.0131 for reciprocal rank ($p = 2.32 \times 10^{-5}$) and -0.0107 for P@1 ($p = 1.48 \times 10^{-4}$).

Within-frequency comparison. As a complementary check, we divide words into low-, mid-, and high-frequency tertiles within each language and compare retrieval performance across fragmentation levels. Across all three frequency tertiles, retrieval performance declines sharply as target words require more tokenizer pieces. Together with the regression results, this indicates that the fragmentation–retrieval relationship is not explained solely by corpus frequency.

We nevertheless interpret this evidence as associational rather than causal. The regression isolates frequency as an important observable confounder, but it does not establish that fragmentation alone causes weaker lexical retrieval. The intervention experiments in Sections 5 and 6 provide evidence on the effects of reducing fragmentation at the tokenizer and downstream levels.

Algorithm 1 LIFT: Learner-grounded Injection for Fair Tokenization

Require: Tokenizer T, learner word lists $\{W_\ell\}_{\ell \in \mathcal{L}}$, budget B
Ensure: Injected vocabulary $\mathcal{S}$
1: **for** each language $\ell \in \mathcal{L}$ **do**
2: Construct fragmented candidates $\mathcal{C}_\ell \leftarrow \{\, w \in W_\ell : |T(w)| > 1 \,\}$
3: Compute unmet burden $U_\ell \leftarrow \sum_{w \in \mathcal{C}_\ell} (|T(w)| - 1)$
4: **end for**
5: Form global candidates $\mathcal{G} \leftarrow \bigcup_{\ell \in \mathcal{L}} \mathcal{C}_\ell$
6: Define language-specific burdens $b_\ell(u)$ for all $\ell \in \mathcal{L}$ and $u \in \mathcal{G}$ as in Eq. (5)
7: Let $\mathcal{L}^+ \leftarrow \{\, \ell \in \mathcal{L} : U_\ell > 0 \,\}$
8: Solve the maximin problem in Eq. (7) to obtain $\mathbf{x}^\star$
9: Return $\mathcal{S} \leftarrow \{\, u \in \mathcal{G} : x_u^\star = 1 \,\}$

Additive family		Non-additive family	
Operator	**Definition**	**Operator**	**Definition**
Sum	$v_{\text{sum}}(w) = \sum_{j=1}^{k} e(t_j)$	Max	$v_{\max}(w)[\ell] = \max_{1 \le j \le k} e(t_j)[\ell]$
Mean	$v_{\text{mean}}(w) = \frac{1}{k} \sum_{j=1}^{k} e(t_j)$	Product	$v_{\text{prod}}(w)[\ell] = \prod_{j=1}^{k} e(t_j)[\ell]$

Table 8: Composition operators used to construct word-level vectors from token embeddings. The additive family contrasts token-count accumulation (SUM) with length-normalized aggregation (MEAN), while the non-additive family contrasts dimension-wise selection (MAX) with multiplicative interaction (PRODUCT).

E PROOF OF THE LIFT GUARANTEE

Let $t^\star$ denote the optimum of Eq. (7), and let $\mathbf{x}^\star$ be any optimal solution. By the first constraint in Eq. (7), for every language $\ell \in \mathcal{L}^+$,

$$\sum_{u \in \mathcal{G}} x_u^\star \, b_\ell(u) \ge t^\star U_\ell.$$

By the definition of $I_\ell(\mathbf{x})$ in Eq. (5), the left-hand side is the total burden removed for language ℓ under the selected injected vocabulary. Hence,

$$I_\ell(\mathbf{x}^\star) \ge t^\star U_\ell \qquad \forall \ell \in \mathcal{L}^+.$$

Since $\mathcal{L}^+ = \{\, \ell \in \mathcal{L} : U_\ell > 0 \,\}$, division by U_ℓ is valid for every $\ell \in \mathcal{L}^+$. Therefore,

$$\frac{I_\ell(\mathbf{x}^\star)}{U_\ell} \ge t^\star \qquad \forall \ell \in \mathcal{L}^+.$$

Using the definition of relative burden reduction in Eq. (6),

$$R_\ell(\mathbf{x}^\star) = \frac{I_\ell(\mathbf{x}^\star)}{U_\ell},$$

we obtain

$$R_\ell(\mathbf{x}^\star) \ge t^\star \qquad \forall \ell \in \mathcal{L}^+.$$

Thus, every language with non-zero unmet burden receives at least a $t^\star$ fraction reduction of its initial burden under any optimal LIFT solution.

F DOWNSTREAM EXPERIMENTAL SETUP

We evaluate whether multilingual continued pretraining and tokenizer adaptation improve downstream multilingual behavior for a mid-scale open model. Our controlled comparison starts from Gemma-3

4B (Team et al., 2025) and consists of three full continued-pretraining runs: (*i*) CPT, which keeps the original tokenizer and updates only the model weights; (*ii*) CPT + LIFT-3K; and (*iii*) CPT + LIFT-5K. This design separates weight adaptation from tokenizer adaptation: CPT measures continued pretraining alone, while the LIFT variants measure the additional effect of vocabulary injection under the same training recipe.

Languages, training mix, and pretraining data. The downstream study uses 12 target languages: Arabic, Bengali, Burmese, French, German, Hindi, Indonesian, Italian, Japanese, Korean, Portuguese, and Yoruba. For target-language continued pretraining, we sample from FineWeb2,[6] a large multilingual pretraining dataset extending FineWeb to more than 1,000 languages (Penedo et al., 2025). For English replay, we use FineWeb-Edu,[7] a 1.3T-token educational subset of FineWeb that provides high-quality English text during continued pretraining.

Each of the three continued-pretraining runs uses a fixed 2B-token budget,[8] with 20% English replay and 80% target-language data. Each run makes a single pass over the same sampled-text manifest, with the multilingual portion divided equally across the 12 target languages. English replay serves as a stability anchor during multilingual adaptation. Using the same manifest across CPT and both LIFT variants keeps data exposure fixed and isolates the effect of vocabulary injection.

We use the same sampled corpora for the LEXCORE overlap analysis, with language-specific word segmentation for Japanese, Thai, and Burmese to avoid whitespace-tokenization artifacts in languages that do not consistently mark word boundaries with spaces (Appendix G).

Benchmarks and metrics. We evaluate multilingual question answering and machine translation. For QA, we use Global-MMLU-Lite (Singh et al., 2025), a multilingual and human-annotated benchmark suite, which avoids relying on machine-translated test items.[9] Translation is evaluated bidirectionally, English→X and X→English, on FLORES+ (NLLB Team et al., 2024), a multilingual parallel benchmark used in our evaluation pipeline. We report QA accuracy and chrF (Popović, 2015) for both translation directions, along with average chrF. We use chrF because its character n-gram formulation is largely independent of tokenization and word segmentation, making it suitable for multilingual evaluation across diverse morphologies, scripts, and tokenizer designs; prior work reports that chrF correlates well with human judgments (Popović, 2015). We compute chrF with SacreBLEU for reproducibility and comparability with standard MT evaluation practice (Post, 2018).

We also report Information Parity (IP) (Tsvetkov & Kipnis, 2024) as an aggregate multilingual support metric. IP is an information-theoretic measure that compares the model's negative log-likelihood for text in a target language against the negative log-likelihood of the corresponding English text. We compute IP on parallel FLORES+ data, using English as the reference language. Unlike perplexity, IP is less sensitive to tokenizer variation across languages and models, making it appropriate for comparing systems with different tokenizers. Finally, we report decoding efficiency using tokens per second and average words per second. We emphasize words per second in the tradeoff analysis because it better reflects language-facing decoding efficiency when tokenizers differ.

Baselines and ablations. Our primary scientific comparison is among the Gemma-3 4B variants, which share the same base architecture, training budget, language mix, and optimization settings. To contextualize performance, we also report measured results for open multilingual or multilingual-capable models, including Aya 8B (Dang et al., 2024), Gemma-3 4B (Team et al., 2025), Llama 3.1 8B (Grattafiori et al., 2024), Ministral 3B and 8B[10], and Qwen3 4B and 8B (Yang et al., 2025). These are not size-matched controls; they provide external reference points for interpreting the quality, parity, and efficiency of the adapted Gemma-3 4B variants. The controlled ablations are: (*i*) weight adaptation only, Gemma-3 4B + CPT; (*ii*) moderate vocabulary injection, Gemma-3 4B + CPT + LIFT-3K; and (*iii*) larger vocabulary injection, Gemma-3 4B + CPT + LIFT-5K.

[6] https://huggingface.co/datasets/HuggingFaceFW/fineweb-2

[7] https://huggingface.co/datasets/HuggingFaceFW/fineweb-edu

[8] The token budget is computed using the `gpt-4o` budgeting tokenizer from https://github.com/openai/tiktoken.

[9] https://huggingface.co/datasets/CohereLabs/Global-MMLU-Lite

[10] https://mistral.ai/news/mistral-3

Training configuration. All Gemma-3 4B adaptation runs use the same continued-pretraining recipe. We pack sequences to a context length of 1024 tokens and use a sequence shuffle buffer of 8192 with shards of 4096 packed sequences. Training uses per-device batch size 4, gradient accumulation 16, learning rate 2×10^{-5}, weight decay 0.1, warmup ratio 0.03, bf16 mixed precision, gradient checkpointing, and 4 data workers. The same optimization and evaluation settings are used for CPT, CPT + LIFT-3K, and CPT + LIFT-5K; only the tokenizer changes for the LIFT ablations. A compact summary is given in Table 9.

F.1 Compute and Reproducibility

The main downstream comparison consists of three full-model Gemma-3 4B continued-pretraining runs: CPT, CPT + LIFT-3K, and CPT + LIFT-5K. Each run processes 2B tokens in a single pass over the same sampled-text manifest, with the same 20% English / 80% multilingual mix and optimization settings. Experiments are run on a single NVIDIA GB10 Blackwell GPU with a Grace CPU host and 128GB system memory.

Under this configuration, each 2B-token Gemma-3 4B continued-pretraining run takes approximately 173.2 hours of training wall clock, excluding downstream evaluation and one-time data packing. Runtime may vary with checkpointing overhead, data-loading efficiency, and evaluation cadence.

To support reproducibility, the comparison is controlled at three levels. First, all Gemma-3 4B variants use the same sampled-text manifest, constructed once using the base Gemma-3 4B tokenizer as the reference budgeting tokenizer. Second, all three continued-pretraining runs use the same token budget, sequence length, optimizer settings, and checkpoint schedule. Third, downstream evaluation uses the same language set, metrics, task directions, prompting, and decoding rules. Under this design, CPT isolates weight adaptation alone, while the LIFT variants isolate the additional contribution of tokenizer modification under the same continued-pretraining budget.

F.2 Robustness of Downstream Results

We complement the controlled Gemma-3 4B comparison with paired statistical tests over matched evaluation examples. These analyses assess the robustness of the differences among CPT, LIFT-3K, and LIFT-5K. Table 10 reports relative changes and statistical significance for three pairwise comparisons, with relative changes computed with respect to the second system in each comparison.

Statistical testing. All tests operate on matched evaluation examples. For QA, we use a two-sided exact McNemar test over 6,400 multiple-choice correctness outcomes. For Information Parity, we use a two-sided paired Student's t-test over 24,108 matched FLORES+ language–sentence IP/NLL differences. For translation, we apply paired bootstrap resampling over sentence pairs, recomputing chrF for each resample ($n = 3{,}600$ per direction). Average chrF uses a direction-stratified paired bootstrap with equal weight on the English→X and X→English directional differences.

Results. LIFT-3K improves over CPT on all five quality and parity measures. Its significant relative gains in IP and translation range from **1.61%** to **2.27%**, alongside a positive **1.66%** QA gain. LIFT-5K likewise improves over CPT on every quality measure, with significant gains in IP, English→X chrF, and average chrF. The direct comparison further favors LIFT-3K for IP, X→English chrF, and average chrF, establishing it as the strongest downstream variant under the fixed 2B-token budget.

Vocabulary budget and adaptation data. The 3K and 5K ablations reveal a tradeoff between fragmentation reduction and the training signal available to newly introduced vocabulary. Larger injection budgets reduce more learner-vocabulary fragmentation (Section 5), but also introduce more token representations that must be integrated during continued pretraining.

We initialize each injected token with the sum of its existing subword embeddings, following Section 4, rather than random initialization. This provides a model-native starting point, while continued pretraining adapts the new representations to the multilingual data. Under the fixed 2B-token budget, LIFT-3K yields the strongest downstream results, whereas LIFT-5K reduces fragmentation further but produces smaller quality gains.

Setting	Value
Model and tokenizer	
Base model	Gemma-3 4B (Team et al., 2025)
Tokenizer control	Original Gemma-3 4B tokenizer
Tokenizer ablations	LIFT-3K, LIFT-5K
Training budget	
Main-comparison budget per run	2B tokens
Epoch / pass	1 pass over the locked shared manifest
English replay	20% of total tokens
Target-language mix	80% of total tokens
Target languages	12
Controlled adaptation runs	3
Runtime per run	≈173.2 hours
Optimization	
Context length	1024 tokens
Per-device batch size	4
Gradient accumulation	16
Learning rate	2×10^{-5}
Weight decay	0.1
Warmup ratio	0.03
Precision	bf16
Gradient checkpointing	enabled
Hardware	
GPU setup	single NVIDIA GB10 Blackwell GPU
CPU	NVIDIA GB10 Grace CPU
System memory	128GB LPDDR5X
Operating system	NVIDIA DGX OS 7
Evaluation	
QA benchmark	Global-MMLU-Lite (Singh et al., 2025)
MT benchmark	FLORES+ (NLLB Team et al., 2024)
MT directions	English→X and X→English
Quality metrics	QA accuracy, chrF, IP
Efficiency metrics	tok/s, words/s
Controlled across variants	
Training data	same sampled-text manifest
Optimizer and schedule	held fixed
Context length	held fixed
Batch construction	held fixed
Evaluation pipeline	held fixed

Table 9: **Downstream adaptation configuration.** All Gemma-3 4B variants use the same continued-pretraining recipe except for tokenizer modification in the LIFT runs.

Table 10: Paired comparisons among the controlled Gemma-3 4B variants. Values report relative percentage changes in the first system with respect to the second. "n.s." denotes a non-significant difference.

Comparison	IP	QA	En→X chrF	X→En chrF	Avg. chrF
LIFT-3K vs. CPT	+1.74%, $p < 10^{-20}$	+1.66%, $p = 0.3652$ (n.s.)	+2.27%, $p < 0.01$	+1.61%, $p < 0.05$	+2.04%, $p < 10^{-7}$
LIFT-5K vs. CPT	+1.07%, $p < 10^{-8}$	+1.04%, $p = 0.5714$ (n.s.)	+1.42%, $p < 0.05$	+0.46%, $p = 0.4237$ (n.s.)	+1.02%, $p < 0.01$
LIFT-3K vs. LIFT-5K	+0.66%, $p < 0.001$	+0.62%, $p = 0.7342$ (n.s.)	+0.84%, $p = 0.2301$ (n.s.)	+1.15%, $p < 0.05$	+1.01%, $p < 0.01$

This pattern reveals a vocabulary-budget–adaptation-data relationship: as the injected vocabulary grows, the available continued-pretraining signal is distributed across more newly introduced representations. Prior work on vocabulary expansion similarly emphasizes the importance of initialization and continued pretraining for integrating new vocabulary (Mundra et al., 2024; Yamaguchi et al., 2026). Our results indicate that larger injection budgets require correspondingly greater adaptation exposure to realize their additional fragmentation reduction.

Language	QA	MT directions
Arabic	yes	En↔Ar
Bengali	yes	En↔Bn
Burmese	yes	En↔My
French	yes	En↔Fr
German	yes	En↔De
Hindi	yes	En↔Hi
Indonesian	yes	En↔Id
Italian	yes	En↔It
Japanese	yes	En↔Ja
Korean	yes	En↔Ko
Portuguese	yes	En↔Pt
Yoruba	yes	En↔Yo

Table 11: **Downstream language set and task directions.**

Table 12: **Prompting and decoding configuration for downstream evaluation.**

Task	Prompt template	Decoding and prediction rule

QA	`You are taking a multiple-choice exam.`
	`Only choose the single best answer.`
	`[QUESTION] question`
	`[OPTIONS]`
	`A) option_a`
	`B) option_b`
	`C) option_c`
	`D) option_d`
	`Answer:`
		Greedy decoding with `do_sample=False`, `temperature=0.0`, `top_p=1.0`, no beam search, `max_new_tokens=16`, and batch size 8. Predictions follow `gen_match_then_loglik`: if the generated text matches an answer option, the matched option is used; otherwise, labels A–D are scored by log-likelihood and the highest-scoring label is selected.

Translation	`You are a professional translator.`
	`Translate the text from {src_lang} to {tgt_lang}.`
	`Preserve meaning, named entities, numbers, and punctuation.`
	`Output only the translation, with no explanation.`
	`Source ({src_lang}): source_text`
	`Translation ({tgt_lang}):`
		Greedy decoding with `do_sample=False`, `temperature=0.0`, `top_p=1.0`, `top_k=None`, no beam search, and batch size 8. Translation uses a dynamic length budget: `max_new_tokens` $= \min(256, \max(32, \lceil 2.0 \cdot$ `source_token_len` $\rceil + 8))$. Output text is lightly normalized to remove trivial wrappers such as fenced blocks or prefixes like `Translation:`.

Under the fixed 2B-token budget, LIFT-3K is better matched to the available data, while the 5K expansion likely requires additional continued pretraining to fully integrate its larger injected vocabulary.

F.3 Benchmark Details

Table 11 summarizes the language set and task directions used in downstream evaluation. QA is evaluated on the same 12 target languages used for continued pretraining. Translation is evaluated in both English→X and X→English directions for the same language set.

F.4 Prompting and Decoding Details

For completeness, Table 12 summarizes prompting and decoding for downstream QA and translation. For tokenizers with a chat template, we wrap the task prompt as a single user turn and enable the generation prompt; otherwise, we use the raw prompt string.

Table 13: Single-Token Retention Rate (STRR, %) for foundational learner vocabulary across 53 languages and 10 tokenizers. Values are rounded to one decimal, and darker cells indicate higher single-token retention. The heatmap shows that lexical integrity is highly uneven across languages and tokenizers: Latin-script and high-resource languages often receive stronger whole-word coverage, while several non-Latin and low-resource languages remain heavily fragmented. Gemma-3 provides the broadest coverage overall, with particularly strong retention for languages such as Japanese, French, German, Thai, and Hindi, whereas several tokenizers assign near-zero single-token coverage to languages such as Bengali, Gujarati, Kannada, Tamil, Telugu, and Sinhala. These patterns show that tokenizer vocabulary allocation is not merely a matter of vocabulary size, but reflects language- and script-specific allocation decisions that shape multilingual lexical access before model inference.

Tokenizers	Arabic (arb)	Basque (eus)	Bengali (ben)	Burmese (mya)	Cebuano (ceb)	Chichewa (nya)	Danish (dan)	Dutch (nld)	Filipino (fil)	Finnish (fin)	French (fra)	German (deu)	Greek (ell)	Gujarati (guj)	Haitian (hat)	Hausa (hau)	Hindi (hin)	Hungarian (hun)	Igbo (ibo)	Indonesian (ind)	Irish (gle)	Italian (ita)	Japanese (jpn)	Javanese (jav)	Kannada (kan)	Korean (kor)	Kyrgyz (kir)
openai/GPT-4	0.9	17.2	0.0	0.3	3.4	3.7	21.8	21.2	14.2	7.0	27.0	22.8	1.0	0.0	18.5	28.9	0.2	10.7	20.7	14.2	9.9	15.1	8.2	17.2	0.0	7.6	2.5
openai/GPT-5	15.9	19.9	5.6	3.3	10.6	6.6	32.3	41.2	18.2	10.5	32.1	42.8	4.1	5.5	23.8	34.1	9.2	15.1	32.8	20.9	13.9	19.2	26.3	21.8	2.0	23.8	12.1
openai/gpt-oss-120b	15.9	19.9	5.6	3.3	10.6	6.6	32.3	41.2	18.2	10.5	32.1	42.8	4.1	5.5	23.8	34.1	9.2	15.1	32.8	20.9	13.9	19.2	26.3	21.8	2.0	23.8	12.1
CohereLabs/aya-expanse-32b	14.4	21.4	0.0	0.4	14.0	5.6	30.5	42.4	17.8	10.6	48.4	42.9	7.4	0.0	25.1	35.8	4.5	14.5	29.1	26.8	15.9	28.4	58.5	25.0	0.0	36.2	9.8
mistralai/Ministral-3-14B	19.5	17.9	1.3	1.4	11.4	4.2	25.1	25.4	14.4	8.4	29.0	34.1	3.5	0.8	19.9	29.5	4.7	18.2	23.4	16.2	11.5	16.6	28.3	16.8	0.9	23.1	7.3
meta-llama/Llama-3.1-70B	9.3	17.4	0.0	0.3	8.1	4.4	22.3	21.5	14.3	7.6	27.0	23.0	4.7	0.0	19.8	29.4	2.7	11.7	23.3	14.9	10.8	15.2	28.8	18.1	0.0	23.0	5.2
Qwen/Qwen3-32B	28.2	17.2	0.1	0.9	9.8	3.7	21.8	21.2	14.3	7.0	28.8	23.9	1.0	0.4	18.6	28.9	0.3	11.3	20.7	14.2	10.8	15.3	43.0	17.2	0.5	23.7	5.6
google/gemma-3-27b	14.2	22.0	28.8	9.5	13.9	5.7	33.1	34.9	19.0	10.4	43.8	44.0	4.4	6.0	25.6	35.2	34.1	17.1	29.3	27.8	14.4	27.1	66.0	24.9	2.2	30.4	9.6
allenai/Olmo-3-1125-32B	0.9	17.2	0.0	0.3	3.4	3.7	21.8	21.2	14.2	7.0	27.0	22.8	1.0	0.0	18.5	28.9	0.2	10.6	20.7	14.2	9.9	15.1	8.2	17.2	0.0	7.6	2.5
deepseek-ai/DeepSeek-V3.2	9.5	17.7	1.9	0.9	9.9	3.8	23.3	23.9	13.5	7.2	27.6	25.9	3.1	0.3	19.2	30.0	1.2	12.8	23.3	16.2	10.9	14.6	35.2	17.2	0.3	15.6	4.5

Tokenizers	Lithuanian (lit)	Malayalam (mal)	Marathi (mar)	Persian (fas)	Polish (pol)	Portuguese (por)	Punjabi (pan)	Russian (rus)	Serbian (srp)	Shona (sna)	Sindhi (snd)	Sinhala (sin)	Somali (som)	Spanish (spa)	Sundanese (sun)	Swedish (swe)	Tamil (tam)	Telugu (tel)	Thai (tha)	Turkish (tur)	Ukrainian (ukr)	Urdu (urd)	Vietnamese (vie)	Xhosa (xho)	Yoruba (yor)	Zulu (zul)
openai/GPT-4	4.4	0.6	0.0	0.5	9.7	16.0	0.5	3.1	3.2	5.8	0.7	0.4	11.2	16.4	9.9	22.0	0.0	0.0	0.1	11.3	1.5	1.1	8.9	3.5	23.9	6.6
openai/GPT-5	6.6	1.8	5.8	18.0	12.4	22.6	2.4	10.7	11.3	9.1	14.7	1.5	14.6	21.8	14.7	30.5	1.3	1.7	23.2	15.6	9.7	16.7	10.0	6.9	35.8	9.6
openai/gpt-oss-120b	6.6	1.8	5.8	18.0	12.4	22.6	2.4	10.7	11.3	9.1	14.7	1.5	14.6	21.8	14.7	30.5	1.3	1.7	23.2	15.6	9.7	16.7	10.0	6.9	35.8	9.6
CohereLabs/aya-expanse-32b	7.4	0.6	1.8	25.9	15.5	29.0	0.5	15.1	14.0	10.2	13.2	0.4	15.1	25.4	17.8	29.3	0.1	0.0	0.1	18.9	13.9	14.2	21.7	5.1	30.0	7.4
mistralai/Ministral-3-14B	5.8	0.7	3.1	17.2	10.8	17.6	1.1	9.6	10.6	6.8	13.2	0.4	11.5	17.0	11.2	25.2	0.4	1.0	8.3	13.5	9.0	13.3	11.2	3.8	26.0	5.9
meta-llama/Llama-3.1-70B	4.9	0.6	0.8	14.8	10.2	16.3	0.5	8.2	8.6	6.1	10.1	0.4	11.2	16.6	10.5	22.5	0.0	0.0	11.4	14.8	9.2	9.2	9.7	3.7	25.0	6.7
Qwen/Qwen3-32B	4.7	0.8	0.0	13.6	10.7	16.5	1.0	10.1	9.2	5.9	13.9	0.9	11.2	16.6	10.0	22.1	0.2	0.5	18.0	12.9	7.3	10.8	11.5	3.5	24.2	6.6
google/gemma-3-27b	7.9	3.2	22.4	20.2	14.1	27.6	1.9	14.6	13.1	9.3	13.2	2.5	15.3	26.9	18.1	32.2	10.1	3.5	40.9	17.1	11.9	19.2	18.0	5.6	29.9	7.7
allenai/Olmo-3-1125-32B	4.4	0.6	0.0	0.5	9.7	16.0	0.5	3.1	3.2	5.8	0.7	0.4	11.2	16.4	9.9	22.0	0.0	0.0	0.1	11.3	1.5	1.1	8.9	3.5	23.9	6.6
deepseek-ai/DeepSeek-V3.2	5.2	0.5	1.0	12.0	10.6	14.0	0.5	8.9	8.8	5.7	8.6	0.9	11.3	13.1	10.4	23.2	0.3	0.2	32.6	11.2	7.6	6.9	9.7	3.5	25.1	5.7

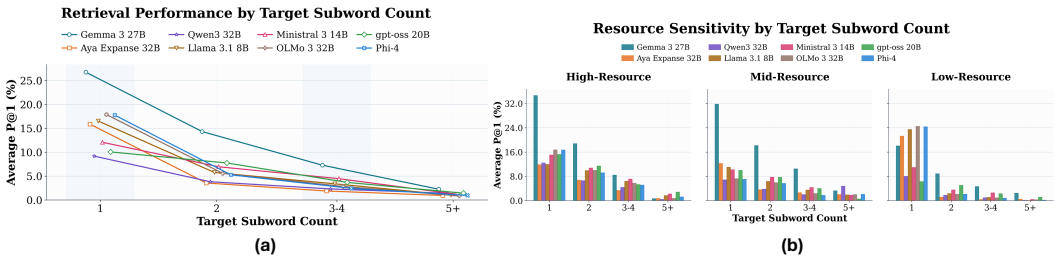


Figure 5: **Subword fragmentation is strongly associated with degraded multilingual lexical retrieval.** (a) Average P@1 across eight representative models as a function of target subword count ('1', '2', '3–4', '5+'). Retrieval performance generally decreases as the target requires more subword pieces, indicating that lexical grounding becomes progressively more difficult under greater compositional load. This mirrors the MRR analysis in Figure 3 while showing that the same effect is also visible in top-1 retrieval accuracy. (b) The same analysis stratified by language resource class ('High', 'Mid', 'Low'). The same trend is observed in every stratum, with the largest degradation typically appearing in lower-resource languages.

G Language-Specific Word Segmentation

Several languages in our evaluation set do not reliably mark word boundaries with whitespace. This matters for any preprocessing step that treats whitespace-delimited strings as word types, such as filtering single-word entries, computing word-level overlap, or estimating word-level fragmentation

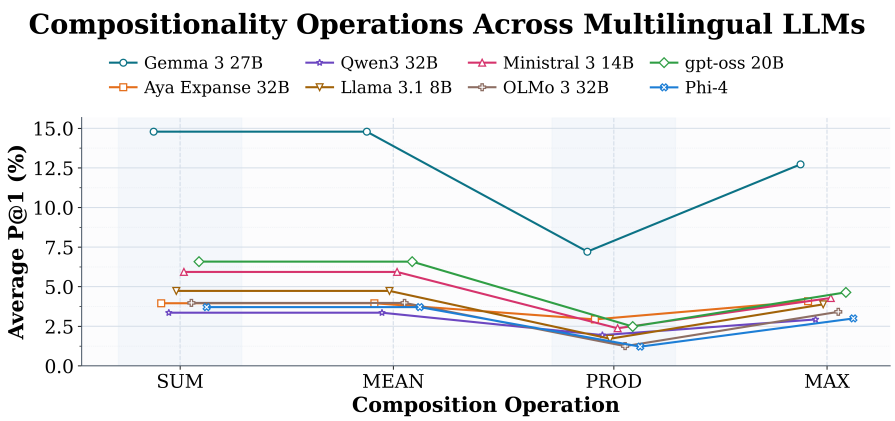


Figure 6: Composition-operation sensitivity across multilingual LLM embeddings, measured by average P@1 over 53 evaluation languages. The ranking largely mirrors the MRR results: SUM is slightly stronger in the main MRR analysis (Figure 2), while MEAN remains very close at top rank. The additive operators consistently outperform the non-additive ones, with PROD the weakest choice and MAX typically below SUM and MEAN. The agreement between P@1 and MRR suggests that the preference for additive composition, especially SUM for model-native initialization, is robust across both top-rank and full-ranking evaluations.

statistics. In particular, Japanese, Thai, and Burmese can contain multiword expressions without spaces; therefore, a "no-space" heuristic is not a reliable proxy for "single word."

To avoid misclassifying phrases as word types, we apply language-specific word segmentation for these languages before word-level filtering and corpus-level word statistics. For Japanese, we use `fugashi` with `unidic-lite`. `fugashi` provides a Python interface to MeCab-style Japanese morphological analysis, while `unidic-lite` supplies a pip-installable UniDic dictionary.[11] For Thai, we use PyThaiNLP's `word_tokenize(..., engine="newmm")`. The `newmm` engine is documented as PyThaiNLP's default word-tokenization engine and implements dictionary-based maximal matching constrained by Thai character-cluster boundaries.[12] For Burmese, we use `pyidaungsu`, which provides Myanmar-language processing utilities and supports word-level tokenization for Burmese.[13]

This language-specific segmentation is used only to make word-level preprocessing more linguistically appropriate for scripts without consistent whitespace word boundaries. It does not change the model tokenizer being evaluated. The tokenizer metrics still measure how each LLM tokenizer segments the original surface forms; the external segmenters are used only to avoid artifacts in auxiliary word-level filtering and corpus statistics. This distinction is important because treating all no-space strings as single words would overestimate lexical coverage for Japanese, Thai, and Burmese and could distort comparisons with languages whose word boundaries are explicitly marked by spaces.

H Composition of the 53-language Evaluation Set

For readability, the grouped analyses in the main text use abbreviated bucket names. In particular, Table 1 uses *IndoE* (Indo-European), *AustN* (Austronesian), *NigerC* (Niger–Congo), and *Drav* (Dravidian), while all remaining lower-frequency family groups are aggregated into *Others*. Likewise, script summaries use *Latn*, *Cyrl*, *Arab*, and *Others*. Table 20 provides the corresponding counts for the full 53-language catalog.

[11] https://pypi.org/project/fugashi/
[12] https://pythainlp.org/docs/4.0/api/tokenize.html
[13] https://pypi.org/project/pyidaungsu/

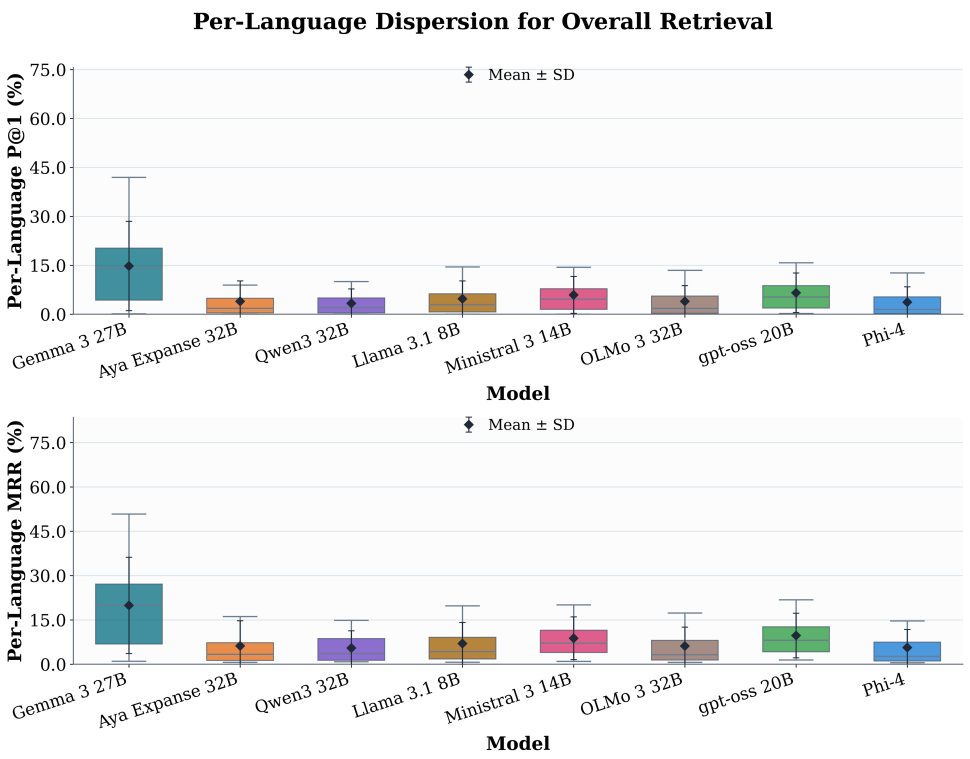
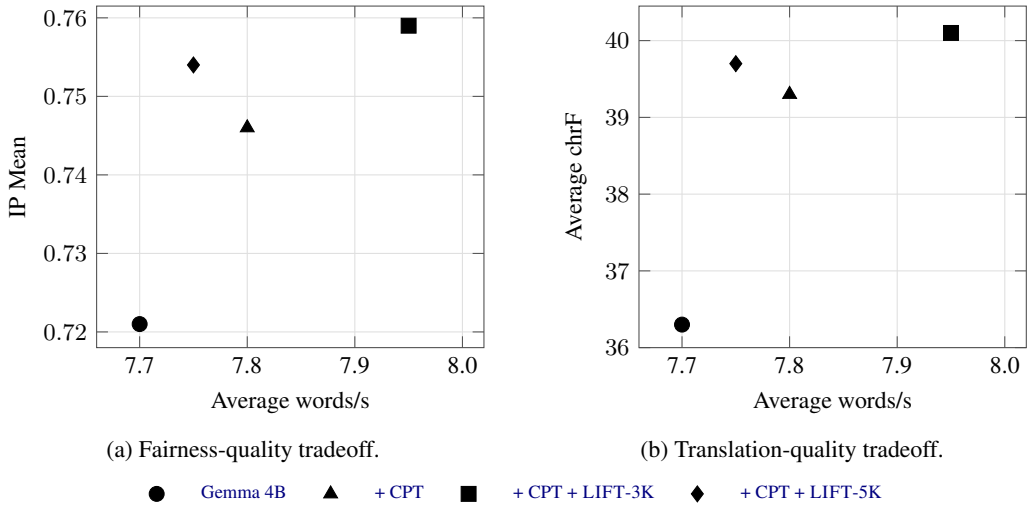


Figure 7: **Average retrieval performance can mask substantial cross-lingual variation.** Boxplots summarize the distribution of language-level retrieval scores over the 53 languages for each model, with diamonds and error bars indicating the mean and one standard deviation. The top panel reports P@1 and the bottom panel reports MRR. Although all models exhibit cross-lingual dispersion, the magnitude of that dispersion varies substantially across architectures, showing that multilingual consistency is a distinct and consequential axis of evaluation. In particular, models with similar aggregate performance can differ markedly in how evenly that performance is distributed across languages, making dispersion informative for assessing multilingual robustness and parity.

I Supplementary Literature Review

Learner vocabulary as a lexical core. Our work draws on second-language learning and vocabulary acquisition, where high-utility foundational vocabulary is central to early comprehension, communicative use, and broader lexical development (Nation, 2001; Webb & Nation, 2017; Schmitt, 2019). Work on language transfer and bilingual lexical processing further shows that early *L2* learning is often mediated by a stronger *L1* lexicon (Odlin, 1989; Kroll & Stewart, 1994). These traditions motivate our learner-vocabulary lens, but they have rarely been connected systematically to multilingual LLM analysis. We make this connection explicit through LexCore, an LLM-facing multilingual learner-vocabulary lexicon used as a compact lexical probe for diagnosis and intervention.

Tokenizer inequity in multilingual LLMs. Tokenization is increasingly recognized as a core modeling decision rather than a neutral preprocessing step (Mielke et al., 2021; Alqahtani et al., 2026). Prior work shows that tokenizer design affects monolingual and multilingual model performance (Rust et al., 2021), vocabulary allocation and overlap across languages (Limisiewicz et al., 2023), and compression behavior across scales (Lotz et al., 2025). Recent studies further show that tokenizers can introduce systematic cross-lingual disparities, causing some languages to require substantially more tokens for the same content and thereby increasing cost, latency, and context-window pressure (Ahia et al., 2023; Petrov et al., 2023; Arnett et al., 2025). Our work complements this literature by shifting the unit of analysis from broad corpus-level token cost to foundational learner vocabulary, exposing where multilingual support fails at the lexical core.



(a) Fairness-quality tradeoff.

(b) Translation-quality tradeoff.

Figure 8: **Quality-efficiency tradeoffs in the Gemma-3 4B CPT comparison.** The two panels place the original Gemma-3 4B and the three Gemma-3 4B CPT variants on a shared average decoding-efficiency axis. The CPT control uses the same tokenizer as the original Gemma-3 4B, so its movement reflects weight adaptation alone; the LIFT variants add tokenizer modification on top of continued pretraining. Continued pretraining improves both parity and translation quality, and LIFT-3K extends this frontier further while retaining most of the efficiency of CPT alone. By contrast, LIFT-5K yields smaller additional quality gains while incurring a somewhat larger efficiency cost.

Vocabulary expansion and tokenizer adaptation. Several approaches seek to mitigate tokenizer mismatch by redesigning tokenizers, improving vocabulary allocation, or expanding vocabularies for target languages. Parity-aware BPE modifies the BPE objective to improve cross-lingual fairness during tokenizer training (Foroutan et al., 2026). Vocabulary-expansion methods instead adapt existing models by adding new tokens, initializing their embeddings, and continuing training for a target language or language set (Kim et al., 2024; Mundra et al., 2024; Yamaguchi et al., 2026). These works are closely related to our intervention, but they typically focus on language adaptation, initialization strategies, or tokenizer-training objectives. In contrast, LIFT is learner-grounded and tokenizer-native: it diagnoses fragmented learner words under an existing tokenizer, selects shared surface forms under a global budget, and optimizes for fair relative burden reduction without training a new tokenizer from scratch.

Subword composition and lexical grounding. A parallel line of work studies how subword units compose into word-level representations. Prior work has examined subword-level composition for word embeddings (Li et al., 2018), lexically grounded subword segmentation (Libovický & Helcl, 2024), and how LLMs compose subword information across layers (Peng et al., 2025). Related work further shows that language-model embeddings exhibit shared geometric structure across models (Lee et al., 2025; Kulkarni et al., 2026). Complementary studies suggest that multilingual LLM behavior can remain English-centric in inference or internal representation (Etxaniz et al., 2024; Schut et al., 2025). Our work brings these threads together by asking whether tokenizer-induced fragmentation of learner vocabulary weakens embedding-only bilingual retrieval, and by using the resulting compositional evidence to motivate sum-based initialization for injected tokens.

J Implications for Multilingual Tokenizer Design

Standard subword tokenizers such as BPE are learned mainly from corpus statistics, so they naturally favor languages that dominate the training data. Simply adding more multilingual data does not fully solve this problem. As the number of languages grows, it becomes harder to control language coverage fairly because high-resource languages still contribute much more data.

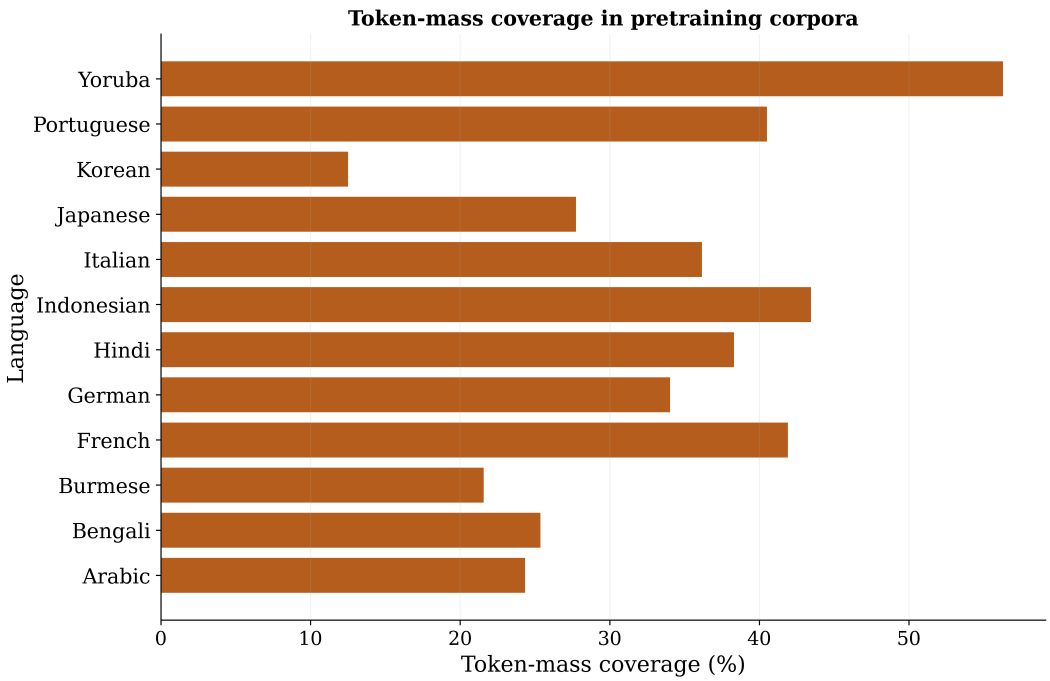


Figure 9: **Token-mass coverage of the 1K translation lexicons across the 12 training-evaluation languages.** For each language, we report the fraction of running corpus token mass in the pretraining data accounted for by the corresponding 1K translation lexicon. Together with Figure 4, the results show that these lexicons are not only well represented among high-frequency vocabulary types, but also cover a substantial portion of the token mass in pretraining corpora. However, token-mass coverage exhibits greater cross-language variation than type-level overlap, showing that strong lexical overlap does not imply uniform concentration in running text.

At the same time, moving too far away from the original BPE vocabulary is also not ideal. Unconstrained vocabulary expansion can scale poorly as more languages are added because every new token increases the embedding and output-softmax size. Training a new tokenizer also typically means retraining the model from scratch, which is very costly.

Our motivation is to address this gap with a human-centered and pedagogically motivated vocabulary, rather than relying only on model-specific corpus statistics. In Section 3, STRR identifies foundational learner words that are poorly represented by the existing tokenizer. In Section 5, LIFT then selects which of these words to inject fairly and under a fixed vocabulary budget, so high-resource languages do not simply receive most of the added capacity. Based on the findings in Section 4, we initialize each new token using the sum of its existing subword embeddings rather than random initialization.

This lets us keep the existing tokenizer and pretrained model, carefully inject a limited number of useful tokens, and continue pretraining instead of retraining the tokenizer and model from scratch. This is especially important for improving multilingual coverage without making adaptation prohibitively expensive. More broadly, the framework separates three design decisions: which lexical units are poorly represented, how a fixed vocabulary budget should be distributed across languages, and how newly introduced tokens should be initialized.

The same idea can also guide tokenizer design before pretraining. STRR can be used together with standard corpus-compression objectives (Section 3), while LIFT can help distribute a fixed vocabulary budget more fairly across languages (Section 5). In this setting, learner-oriented vocabulary complements rather than replaces corpus statistics, providing an explicit coverage objective alongside corpus-driven tokenization.

Our current LEXCORE already covers 105 languages, and the framework can support all of them and be extended to additional languages or broader vocabulary sets, including those suggested by the

pretraining-overlap results in Figures 4 and 9. Broader lexical inventories could therefore combine learner-oriented vocabulary with corpus-derived candidates while retaining an explicit budget for cross-lingual allocation. More generally, our findings suggest treating multilingual lexical coverage as an explicit allocation problem rather than as a by-product of corpus frequency alone.

K LIMITATIONS

Evaluation scope. Our evaluation is intentionally staged from broad diagnosis to controlled intervention. LEXCORE covers 105 languages, while the tokenizer analysis, embedding-only retrieval, and intrinsic LIFT evaluation span 53 languages across 10 tokenizer/model families and multiple corpora. Downstream evaluation uses the 12-language intersection of LEXCORE, human-annotated Global-MMLU-Lite, and FLORES+, enabling matched multilingual QA and bidirectional translation without machine-translated QA data. This final stage tests whether the earlier tokenizer- and representation-level findings translate into downstream gains under a controlled setting. Broader language coverage, dialectal varieties, non-English-centric translation, and open-ended generation would further characterize generality.

Controlled continued pretraining. The downstream study compares three full Gemma-3 4B continued-pretraining runs: CPT, CPT + LIFT-3K, and CPT + LIFT-5K. All runs use the same sampled-text manifest, 2B-token budget, language mix, optimization recipe, and evaluation pipeline, isolating the contribution of vocabulary injection. Paired tests over matched examples provide additional robustness evidence (Appendix F.2). Repeated runs across seeds, model families, and scales would provide complementary evidence on training-level variability and transfer.

Vocabulary budget and adaptation data. LIFT treats vocabulary expansion as an explicit allocation problem because injected tokens enlarge the embedding and output-vocabulary matrices and must be integrated through continued pretraining. Under the same 2B-token budget, LIFT-3K gives the strongest downstream results, while LIFT-5K removes more fragmentation but yields smaller quality gains. This suggests that the appropriate injection budget should scale with the available adaptation data. Larger training budgets therefore provide a natural setting for larger vocabulary expansions, while pruning unused, rare, or undertrained tokens offers a complementary balancing strategy (Cognetta et al., 2024).

Wordhood and segmentation. LEXCORE uses foundational learner vocabulary as a consistent lexical unit across languages, although word boundaries vary across writing systems and morphological traditions (Haspelmath, 2011). We therefore use language-specific segmentation for Japanese, Thai, and Burmese in auxiliary filtering and corpus statistics, while applying the evaluated LLM tokenizers to the original surface forms. Morphemes, compounds, and multiword constructions could provide complementary units where orthographic words are less informative.

Composition and initialization. Our lexical-grounding experiments identify additive composition as a robust reconstruction strategy, with SUM providing a slight aggregate advantage and a natural model-native initialization for LIFT. We therefore use summed subword embeddings in the downstream intervention. Learned, frequency-aware, or architecture-specific initialization strategies could provide complementary gains.

Scaling the lexical inventory. LEXCORE provides a learner-grounded foundation across 105 languages, not an endpoint for multilingual vocabulary design. Figure 4 and Figure 9 show that these learner lexicons align strongly with pretraining corpora at both the frequent-type and token-mass levels, providing a principled basis for identifying additional corpus-derived lexical candidates. Larger models and training regimes can therefore use LEXCORE as an initial human-centered coverage objective, augment it with corpus evidence, and scale the injection budget with available vocabulary capacity and continued-pretraining data. The same principle extends beyond the current 105 languages by combining validated learner-oriented inventories with language-specific corpus statistics. More broadly, our framework treats multilingual lexical coverage as a deliberate allocation problem rather than a by-product of corpus frequency alone.

Table 14: Token-length distribution of foundational learner vocabulary across 53 languages and 10 tokenizers. Each tokenizer is shown as a three-row block reporting STRR, the fraction of words split into exactly two tokens ($P[=2]$), and the fraction split into three or more tokens ($P[\geq 3]$); darker cells indicate larger values. Low STRR is often accompanied by severe multi-piece fragmentation, especially for non-Latin and low-resource languages. Newer tokenizers reduce fragmentation for some languages, with Gemma-3 showing the broadest gains, but substantial $P[\geq 3]$ mass remains across scripts and families. These results show that tokenizer inequity concerns not only whether words are split, but how severely they are fragmented.

Tokenizers	Metric	Arabic (arb)	Basque (eus)	Bengali (ben)	Burmese (mya)	Cebuano (ceb)	Chichewa (nya)	Danish (dan)	Dutch (nld)	Filipino (fil)	Finnish (fin)	French (fra)	German (deu)	Greek (ell)	Gujarati (guj)	Haitian (hat)	Hausa (hau)	Hindi (hin)	Hungarian (hun)	Igbo (ibo)	Indonesian (ind)	Irish (gle)	Italian (ita)	Japanese (jpn)	Javanese (jav)	Kannada (kan)	Korean (kor)	Kyrgyz (kir)
GPT-4	STRR	0.9	17.2	0.0	0.3	3.4	3.7	21.8	21.2	14.2	7.0	27.0	22.8	1.0	0.0	18.5	28.9	0.2	10.7	20.7	14.2	9.9	15.1	8.2	17.2	0.0	7.6	2.5
	$P[=2]$	8.5	37.6	0.6	1.1	16.4	31.5	47.7	48.3	36.2	40.9	50.1	53.0	2.8	0.7	47.9	42.4	2.6	37.1	37.9	54.0	39.7	53.4	25.2	48.8	0.7	31.3	12.8
	$P[\geq 3]$	90.6	45.3	99.4	98.6	80.2	64.8	30.5	30.5	49.6	52.1	22.9	24.3	96.2	99.3	33.6	28.7	97.2	52.2	41.4	31.8	50.4	31.4	66.6	34.0	99.3	61.1	84.7
GPT-5	STRR	15.9	19.9	5.6	3.3	10.6	6.6	32.3	41.2	18.2	10.5	32.1	42.8	4.1	5.5	23.8	34.1	9.2	15.1	32.8	20.9	13.9	19.2	26.3	21.8	2.0	23.8	12.1
	$P[=2]$	63.4	47.0	40.0	17.5	39.0	48.7	50.5	45.0	51.2	53.9	55.1	46.1	22.8	38.6	59.4	52.0	49.2	45.1	43.5	60.3	48.1	57.5	46.1	56.9	23.7	62.4	43.3
	$P[\geq 3]$	20.7	33.1	54.3	79.2	50.4	44.7	17.2	13.8	30.6	35.5	12.8	11.1	73.1	55.9	16.8	13.9	41.5	39.9	23.7	18.8	38.0	23.3	27.6	21.4	74.3	13.8	44.7
gpt-oss-120b	STRR	15.9	19.9	5.6	3.3	10.6	6.6	32.3	41.2	18.2	10.5	32.1	42.8	4.1	5.5	23.8	34.1	9.2	15.1	32.8	20.9	13.9	19.2	26.3	21.8	2.0	23.8	12.1
	$P[=2]$	63.4	47.0	40.0	17.5	39.0	48.7	50.5	45.0	51.2	53.9	55.1	46.1	22.8	38.6	59.4	52.0	49.2	45.1	43.5	60.3	48.1	57.5	46.1	56.9	23.7	62.4	43.3
	$P[\geq 3]$	20.7	33.1	54.3	79.2	50.4	44.7	17.2	13.8	30.6	35.5	12.8	11.1	73.1	55.9	16.8	13.9	41.5	39.9	23.7	18.8	38.0	23.3	27.6	21.4	74.3	13.8	44.7
aya-expanse-32b	STRR	14.4	21.4	0.0	0.4	14.0	5.6	30.5	42.4	17.8	10.6	48.4	42.9	7.4	0.0	25.1	35.8	4.5	14.5	29.1	26.8	15.9	28.4	58.5	25.0	0.0	36.2	9.8
	$P[=2]$	62.5	44.1	2.4	1.0	41.9	41.8	43.7	44.9	48.4	46.0	42.5	45.9	35.7	0.7	51.4	46.7	11.5	43.4	32.4	61.7	41.6	56.3	36.5	53.4	0.7	54.5	33.1
	$P[\geq 3]$	23.2	34.5	97.6	98.6	44.1	52.6	25.9	12.7	33.8	43.4	9.2	11.2	56.9	99.3	23.5	17.4	84.0	42.0	38.5	11.5	42.5	15.3	5.0	21.6	99.3	9.3	57.1
Ministral-3-14B	STRR	19.5	17.9	1.3	1.4	11.4	4.2	25.1	25.4	14.4	8.4	29.0	34.1	3.5	0.8	19.9	29.5	4.7	18.2	23.4	16.2	11.5	16.6	28.3	16.8	0.9	23.1	7.3
	$P[=2]$	62.5	44.8	30.8	8.0	39.5	33.5	50.7	50.1	40.9	48.1	52.7	54.0	23.7	13.3	52.8	46.5	38.4	45.8	35.9	57.0	40.0	58.0	46.8	52.0	13.8	65.2	33.4
	$P[\geq 3]$	18.1	37.4	68.0	90.6	49.1	62.3	24.2	24.6	44.7	43.5	18.3	11.9	72.8	85.8	27.3	24.0	56.9	36.1	40.7	26.8	48.6	25.4	24.9	31.2	85.3	11.7	59.2
Llama-3.1-70B	STRR	9.3	17.4	0.0	0.3	8.1	4.4	22.3	21.5	14.3	7.6	27.0	23.0	4.7	0.0	19.8	29.4	2.7	11.7	23.3	14.9	10.8	15.2	28.8	18.1	0.0	23.0	5.2
	$P[=2]$	55.7	37.9	0.6	1.1	30.4	32.7	47.9	48.8	38.9	41.3	50.9	53.4	22.7	0.7	50.3	43.3	27.7	39.1	35.8	54.1	41.2	53.6	52.1	49.0	0.7	62.5	23.9
	$P[\geq 3]$	35.0	44.8	99.4	98.6	61.5	62.9	29.8	29.6	46.9	51.1	22.1	23.7	72.6	99.3	29.9	27.3	69.6	49.2	40.9	31.0	48.0	31.1	19.1	32.9	99.3	14.5	70.9
Qwen3-32B	STRR	28.2	17.2	0.1	0.9	9.8	3.7	21.8	21.2	14.3	7.0	28.8	23.9	1.0	0.4	18.6	28.9	0.3	11.3	20.7	14.2	10.8	15.3	43.0	17.2	0.5	23.7	5.6
	$P[=2]$	46.8	37.8	4.3	0.8	28.5	31.5	47.7	48.4	36.1	41.1	49.4	53.8	2.9	1.1	49.1	43.0	7.7	38.4	38.3	54.0	40.9	53.3	40.8	48.8	0.9	63.5	23.1
	$P[\geq 3]$	25.0	45.1	95.6	98.3	61.7	64.8	30.5	30.4	49.6	51.9	21.8	22.4	96.1	98.4	32.3	28.1	92.0	50.3	41.0	31.8	48.3	31.3	16.2	34.0	98.6	12.8	71.3
gemma-3-27b	STRR	14.2	22.0	28.8	9.5	13.9	5.7	33.1	34.9	19.0	10.4	43.8	44.0	4.4	6.0	25.6	35.2	34.1	17.1	29.3	27.8	14.4	27.1	66.0	24.9	2.2	30.4	9.6
	$P[=2]$	62.7	44.9	51.5	24.8	47.1	41.4	48.0	48.7	50.3	54.0	45.9	45.6	22.1	42.9	53.7	48.0	51.6	49.3	33.1	60.9	44.8	57.1	31.2	55.1	30.6	61.8	39.5
	$P[\geq 3]$	23.2	33.1	19.7	65.8	38.9	52.8	18.9	16.3	30.7	35.5	10.3	10.4	73.5	51.1	20.7	16.7	14.3	33.6	37.6	11.3	40.8	15.7	2.8	19.9	67.2	7.8	50.9
Olmo-3-1125-32B	STRR	0.9	17.2	0.0	0.3	3.4	3.7	21.8	21.2	14.2	7.0	27.0	22.8	1.0	0.0	18.5	28.9	0.2	10.6	20.7	14.2	9.9	15.1	8.2	17.2	0.0	7.6	2.5
	$P[=2]$	8.5	37.6	0.6	1.1	16.4	31.5	47.7	48.3	36.1	40.9	50.1	53.0	2.8	0.7	47.9	42.4	2.6	37.2	37.9	54.0	39.7	53.4	25.2	48.8	0.7	31.3	12.8
	$P[\geq 3]$	90.6	45.3	99.4	98.6	80.2	64.8	30.5	30.5	49.7	52.1	22.9	24.3	96.2	99.3	33.6	28.7	97.2	52.2	41.4	31.8	50.4	31.4	66.6	34.0	99.3	61.1	84.7
DeepSeek-V3.2	STRR	9.5	17.7	1.9	0.9	9.9	3.8	23.3	23.9	13.5	7.2	27.6	25.9	3.1	0.3	19.2	30.0	1.2	12.8	23.3	16.2	10.9	14.6	35.2	17.2	0.3	15.6	4.5
	$P[=2]$	57.2	40.8	30.6	1.6	31.5	32.7	47.4	50.6	42.0	45.2	51.9	54.9	14.2	4.4	51.6	45.3	17.5	42.3	35.6	57.0	40.9	56.7	46.0	50.7	3.0	59.8	26.4
	$P[\geq 3]$	33.2	41.5	67.6	97.5	58.6	63.5	29.3	25.6	44.5	47.6	20.6	19.2	82.8	95.3	29.2	24.7	81.3	44.8	41.2	26.8	48.2	28.7	18.8	32.1	96.7	24.5	69.0

Tokenizers	Metric	Lithuanian (lit)	Malayalam (mal)	Marathi (mar)	Persian (fas)	Polish (pol)	Portuguese (por)	Punjabi (pan)	Russian (rus)	Serbian (srp)	Shona (sna)	Sindhi (snd)	Sinhala (sin)	Somali (som)	Spanish (spa)	Sundanese (sun)	Swedish (swe)	Tamil (tam)	Telugu (tel)	Thai (tha)	Turkish (tur)	Ukrainian (ukr)	Urdu (urd)	Vietnamese (vie)	Xhosa (xho)	Yoruba (yor)	Zulu (zul)
GPT-4	STRR	4.4	0.6	0.0	0.5	9.7	16.0	0.5	3.1	3.2	5.8	0.7	0.4	11.2	16.4	9.9	22.0	0.0	0.0	0.1	11.3	1.5	1.1	8.9	3.5	23.9	6.6
	$P[=2]$	26.5	0.6	1.3	14.7	29.8	57.6	0.5	18.2	18.3	42.0	11.6	0.4	28.8	57.6	51.2	43.6	0.2	0.8	8.3	40.4	12.4	10.8	52.8	33.1	42.6	25.2
	$P[\geq 3]$	69.1	98.8	98.7	84.9	60.5	26.4	99.0	78.7	78.4	52.3	87.7	99.1	60.0	26.0	39.0	34.4	99.8	99.2	91.6	48.3	86.0	88.1	38.3	63.5	33.6	68.2
GPT-5	STRR	6.6	1.8	5.8	18.0	12.4	22.6	2.4	10.7	11.3	9.1	14.7	1.5	14.6	21.8	14.7	30.5	1.3	1.7	23.2	15.6	9.7	16.7	10.0	6.9	35.8	9.6
	$P[=2]$	39.4	17.8	39.8	56.4	38.5	61.7	21.9	44.5	45.6	54.4	54.4	19.5	40.8	62.7	58.7	50.9	19.3	18.3	32.9	50.8	34.9	51.8	83.8	53.6	47.9	41.5
	$P[\geq 3]$	54.0	80.4	54.4	25.7	49.1	15.8	75.8	44.8	43.1	36.5	30.9	79.0	44.6	15.5	26.6	18.6	79.4	80.0	43.9	33.6	55.4	31.5	6.1	39.5	16.3	48.9
gpt-oss-120b	STRR	6.6	1.8	5.8	18.0	12.4	22.6	2.4	10.7	11.3	9.1	14.7	1.5	14.6	21.8	14.7	30.5	1.3	1.7	23.2	15.6	9.7	16.7	10.0	6.9	35.8	9.6
	$P[=2]$	39.4	17.8	39.8	56.4	38.5	61.7	21.9	44.5	45.6	54.4	54.4	19.5	40.8	62.7	58.7	50.9	19.3	18.3	32.9	50.8	34.9	51.8	83.8	53.6	47.9	41.5
	$P[\geq 3]$	54.0	80.4	54.4	25.7	49.1	15.8	75.8	44.8	43.1	36.5	30.9	79.0	44.6	15.5	26.6	18.6	79.4	80.0	43.9	33.6	55.4	31.5	6.1	39.5	16.3	48.9
aya-expanse-32b	STRR	7.4	0.6	1.8	25.9	15.5	29.0	0.5	15.1	14.0	10.2	13.2	0.4	15.1	25.4	17.8	29.3	0.1	0.0	0.1	18.9	13.9	14.2	21.7	5.1	30.0	7.4
	$P[=2]$	35.6	0.6	7.4	55.7	46.6	58.6	0.5	53.3	47.3	49.0	34.4	0.4	36.9	62.1	57.5	44.8	1.1	0.8	6.0	56.3	46.4	34.9	77.0	41.3	46.8	33.7
	$P[\geq 3]$	57.0	98.8	90.9	18.4	37.9	12.4	99.0	31.6	38.7	40.7	52.4	99.1	48.0	12.4	24.8	26.0	98.8	99.2	93.9	24.8	39.7	50.9	1.3	53.6	23.2	59.0
Ministral-3-14B	STRR	5.8	0.7	3.1	17.2	10.8	17.6	1.1	9.6	10.6	6.8	13.2	0.4	11.5	17.0	11.2	25.2	0.4	1.0	8.3	13.5	9.0	13.3	11.2	3.8	26.0	5.9
	$P[=2]$	34.3	7.8	29.3	57.6	38.8	58.3	13.2	43.8	48.0	43.9	35.7	0.0	32.2	60.4	53.4	49.0	11.3	11.1	29.1	48.2	36.5	50.7	82.9	34.4	45.5	26.7
	$P[\geq 3]$	60.0	91.5	67.6	25.2	50.4	24.1	85.6	46.6	41.4	49.3	51.2	99.6	56.2	22.6	35.4	25.9	88.3	87.9	62.6	38.3	54.5	35.9	5.9	61.8	28.5	67.3
Llama-3.1-70B	STRR	4.9	0.6	0.8	14.8	10.2	16.3	0.5	8.2	8.6	6.1	10.1	0.4	11.2	16.6	10.5	22.5	0.0	0.0	11.4	14.8	9.2	9.2	9.7	3.7	25.0	6.7
	$P[=2]$	27.0	0.6	19.4	59.0	30.7	57.8	0.5	40.3	38.2	43.3	33.6	0.4	29.8	58.0	52.7	43.4	0.2	0.8	35.5	50.8	34.7	36.6	85.7	34.0	44.7	25.9
	$P[\geq 3]$	68.1	98.8	79.8	26.2	59.1	25.9	99.0	51.6	53.3	50.7	56.3	99.1	59.1	25.4	36.8	34.1	99.8	99.2	53.0	34.5	56.1	54.2	4.6	62.3	30.3	67.4
Qwen3-32B	STRR	4.7	0.8	0.0	13.6	10.7	16.5	1.0	10.1	9.2	5.9	13.9	0.9	11.2	16.6	10.0	22.1	0.2	0.5	18.0	12.9	7.3	10.8	11.5	3.5	24.2	6.6
	$P[=2]$	27.7	1.1	3.9	41.0	33.9	57.6	0.8	38.9	34.1	41.9	37.0	3.1	28.8	58.0	51.2	44.6	0.9	1.0	25.3	45.0	22.2	29.8	78.4	33.2	45.8	25.2
	$P[\geq 3]$	67.6	98.1	96.1	45.3	55.4	25.9	98.2	51.1	56.7	52.3	49.1	96.0	60.0	25.4	38.9	33.3	98.9	98.5	56.7	42.1	70.5	59.4	10.0	63.4	30.0	68.2
gemma-3-27b	STRR	7.9	3.2	22.4	20.2	14.1	27.6	1.9	14.6	13.1	9.3	13.2	2.5	15.3	26.9	18.1	32.2	10.1	3.5	40.9	17.1	11.9	19.2	18.0	5.6	29.9	7.7
	$P[=2]$	39.6	32.4	53.0	58.3	42.2	59.3	17.4	56.5	50.6	49.1	49.4	27.3	38.2	60.7	57.4	48.7	48.6	37.0	34.8	55.1	45.9	57.0	79.2	41.4	47.0	33.4
	$P[\geq 3]$	52.5	64.4	24.7	21.5	43.7	13.0	80.7	28.9	36.4	41.6	37.5	70.2	46.5	12.4	24.6	19.1	41.3	59.5	24.3	27.8	42.2	23.8	2.8	53.0	23.1	59.0
Olmo-3-1125-32B	STRR	4.4	0.6	0.0	0.5	9.7	16.0	0.5	3.1	3.2	5.8	0.7	0.4	11.2	16.4	9.9	22.0	0.0	0.0	0.1	11.3	1.5	1.1	8.9	3.5	23.9	6.6
	$P[=2]$	26.5	0.6	1.3	14.7	29.8	57.6	0.5	18.2	18.3	42.0	11.6	0.4	28.8	57.6	51.2	43.6	0.2	0.8	5.8	40.4	12.4	10.8	52.8	33.0	42.6	25.2
	$P[\geq 3]$	69.1	98.8	98.7	84.9	60.5	26.4	99.0	78.7	78.4	52.3	87.7	99.1	60.0	26.0	39.0	34.4	99.8	99.2	94.1	48.3	86.0	88.1	38.3	63.6	33.6	68.2
DeepSeek-V3.2	STRR	5.2	0.5	1.0	12.0	10.6	14.0	0.5	8.9	8.8	5.7	8.6	0.9	11.3	13.1	10.4	23.2	0.3	0.2	32.6	11.2	7.6	6.9	9.7	3.5	25.1	5.7
	$P[=2]$	29.1	2.0	12.8	56.8	37.0	60.3	3.5	41.6	43.5	43.5	33.2	3.7	31.1	60.9	53.6	48.6	5.1	2.7	34.6	40.5	30.7	41.5	68.4	34.6	44.0	27.8
	$P[\geq 3]$	65.7	97.5	86.1	31.1	52.3	25.7	96.0	49.5	47.7	50.9	58.2	95.5	57.6	26.0	36.0	28.2	94.6	97.1	32.7	48.3	61.7	51.5	21.9	61.9	30.9	66.4

Language	QA (%)↑	chrF↑ E→X	chrF↑ X→E	Avg.↑ chrF
Gemma 4B + CPT + LIFT-3K				
Arabic	49.7	34.8	43.0	38.9
Bengali	49.1	32.9	41.8	37.4
Burmese	47.6	28.4	39.2	33.8
French	51.2	40.6	49.8	45.2
German	50.6	39.8	48.7	44.3
Hindi	49.5	35.1	43.6	39.4
Indonesian	49.4	37.4	45.5	41.5
Italian	50.8	41.0	48.9	45.0
Japanese	48.6	33.7	42.0	37.9
Korean	48.9	34.9	43.4	39.2
Portuguese	50.3	40.2	48.1	44.2
Yoruba	41.1	33.2	35.2	34.2
Mean	**48.9**	**36.0**	**44.1**	**40.1**

Table 15: **Per-language results for Gemma 4B + CPT + LIFT-3K.** CPT denotes continued pretraining, and QA is reported as percentages. Improvements are broadly distributed across languages rather than concentrated in a small subset, with particularly strong gains for several script-diverse and tokenizer-sensitive languages.

	Quality					Efficiency		
Model	ΔIP (pp)↑	ΔQA (pp)↑	ΔchrF↑ E→X	ΔchrF↑ X→E	ΔAvg.↑ chrF	Δtok/s↑ E→X	Δtok/s↑ X→E	ΔAvg.↑ words/s
Gemma 4B CPT variants vs. original Gemma 4B								
Gemma 4B + CPT	+2.5	+2.0	+3.5	+2.5	+3.0	**-0.1**	**-0.1**	+0.10
Gemma 4B + CPT + LIFT-3K	**+3.8**	**+2.8**	**+4.3**	**+3.2**	**+3.8**	-0.4	-0.4	**+0.25**
Gemma 4B + CPT + LIFT-5K	+3.3	+2.5	+4.0	+2.7	+3.4	-0.7	-0.7	+0.05

Table 16: **Change relative to the original Gemma 4B.** Positive values indicate improvement; negative values indicate slowdown. CPT denotes continued pretraining, and QA and IP are reported in percentage points. All three Gemma 4B CPT variants improve parity, QA, and translation quality over the original model, showing that multilingual continued pretraining is beneficial even before tokenizer modification. The largest quality gains are achieved by LIFT-3K, while the CPT-only control incurs the smallest efficiency cost. The CPT control uses the same tokenizer, vocabulary, and architecture as the original Gemma 4B, so any changes in QA, translation quality, or information parity reflect weight adaptation rather than tokenization changes. In particular, continued pretraining can alter generation quality, sequence length, and stopping behavior even when segmentation is unchanged. Accordingly, the CPT row isolates the effect of multilingual weight adaptation, while the LIFT rows measure the additional effect of tokenizer modification on top of the same continued-pretraining recipe. Boldface marks the strongest Gemma 4B CPT variant in each column.

Table 17: **FLORES+ resource-class breakdown across metrics.** Lower is better for all reported metrics. We report average corpus token count in thousands of tokens (CTC (K)), token premium relative to English derived from CTC (TP), tokenizer fairness Gini over language-level CTC (TFG), and maximum CTC-based token premium (MTP), each computed within the corresponding resource class. TP uses the English-relative definition from the main analysis, TFG is the Gini coefficient over language-level CTC within a resource class, and MTP is the maximum non-English CTC-based token premium within that resource class. The *Greedy* variants are reported as an unconstrained large-vocabulary reference and are therefore excluded from best-value highlighting; *LIFT-k* denotes our maximin MILP-based vocabulary injection with budget k. Best displayed values among non-Greedy systems in each column are boldfaced. Across Gemma-3, Aya-Expanse, and Qwen3, LIFT most consistently improves the more fragmented mid- and low-resource groups, especially by reducing TP, TFG, and MTP under larger budgets. The gains are not uniformly monotonic for every metric or resource class, but the result shows that budgeted learner-grounded injection can reduce worst-case and fairness-sensitive tokenization disparities without resorting to full Greedy expansion.

	FLORES+											
	CTC (K)↓			TP↓			TFG↓			MTP↓		
Tokenizer	**High**	**Mid**	**Low**	**High**	**Mid**	**Low**	**High**	**Mid**	**Low**	**High**	**Mid**	**Low**
Base tokenizers												
Gemma-3	**73.0**	**83.2**	100.2	**0.402**	0.570	0.891	**0.060**	0.089	0.080	0.679	1.176	1.402
GPT-OSS	75.1	88.6	**96.3**	0.462	0.690	0.838	0.074	**0.078**	0.108	0.788	1.129	2.170
Aya-Expanse	78.7	139.9	207.6	0.527	1.655	2.940	0.149	0.320	0.363	2.235	5.382	8.113
Qwen3	95.5	155.1	202.4	0.825	1.880	2.758	0.155	0.258	0.320	3.430	5.106	7.931
OLMo-3	110.3	183.2	232.8	1.154	2.457	3.394	0.193	0.280	0.365	3.775	6.629	10.637
Llama-3.1	87.4	144.9	221.1	0.694	1.736	3.175	0.101	0.287	0.371	1.528	6.631	10.641
Ministral-3	77.5	96.9	146.0	0.456	0.784	1.688	0.068	0.100	0.247	0.795	1.362	11.433
Our injected tokenizers												
Gemma-3 + Greedy (34,809)	82.0	89.6	101.7	0.441	0.545	0.754	0.063	0.075	0.065	0.666	1.032	1.170
Gemma-3 + LIFT-5,000	74.1	83.6	99.8	0.409	0.562	0.865	**0.060**	0.087	0.077	0.665	1.143	1.374
├── + LIFT-10,000	74.9	83.7	99.3	0.419	0.557	0.847	0.061	0.084	0.075	**0.647**	1.124	1.353
├── + LIFT-20,000	77.6	85.7	99.5	0.435	**0.555**	**0.807**	0.061	**0.078**	**0.070**	0.668	**1.063**	**1.289**
Aya-Expanse + Greedy (37,647)	96.4	139.9	169.8	0.658	1.346	1.848	0.154	0.196	0.290	1.929	3.083	5.941
Aya-Expanse + LIFT-5,000	79.1	137.9	204.4	0.522	1.596	2.849	0.142	0.312	0.360	2.106	5.190	7.966
├── + LIFT-10,000	79.3	134.4	200.6	0.518	1.516	2.756	0.136	0.300	0.357	2.015	4.832	7.789
├── + LIFT-20,000	80.8	130.2	191.5	0.516	1.390	2.514	0.123	0.277	0.344	1.820	4.375	7.271
Qwen3 + Greedy (38,846)	107.8	145.0	167.7	0.661	1.177	1.519	0.132	0.191	0.252	1.725	2.594	4.025
Qwen3 + LIFT-5,000	95.6	153.1	199.4	0.786	1.783	2.625	0.148	0.252	0.318	3.202	4.856	7.669
├── + LIFT-10,000	95.8	150.9	196.0	0.781	1.729	2.543	0.141	0.247	0.314	3.073	4.657	7.427
├── + LIFT-20,000	96.7	146.3	187.8	0.727	1.544	2.265	0.130	0.235	0.305	2.741	4.082	6.693

Table 18: **FineWeb2 resource-class breakdown across metrics.** Lower is better for all reported metrics. We report byte-normalized token cost (TPB), subword fertility (SF), tokenizer fairness Gini (TFG), and proportion of continued words (PCW), each computed within the corresponding resource class. Here, TFG is the Gini coefficient over language-level TPB within a resource class. The *Greedy* variants are reported as an unconstrained large-vocabulary reference and are therefore excluded from best-value highlighting; *LIFT-k* denotes our maximin MILP-based vocabulary injection with budget k. Best values among non-Greedy systems in each column are boldfaced. Across Gemma-3, Aya-Expanse, and Qwen3, LIFT consistently reduces subword fertility and continued-word rates as the budget increases, with the clearest gains in mid- and low-resource groups where fragmentation is highest. TPB and TFG improvements are more distribution-dependent, but larger LIFT budgets generally reduce low-resource fragmentation and improve fairness-sensitive metrics without requiring full Greedy expansion.

	FineWeb2											
	TPB↓			SF↓			TFG↓			PCW↓		
Tokenizer	**High**	**Mid**	**Low**	**High**	**Mid**	**Low**	**High**	**Mid**	**Low**	**High**	**Mid**	**Low**
Base tokenizers												
Gemma-3	**0.221**	**0.208**	0.249	2.044	2.288	2.514	0.127	0.178	0.167	0.596	0.652	0.730
GPT-OSS	0.223	0.213	**0.226**	2.112	2.411	2.442	0.122	0.138	0.135	0.632	0.695	0.738
Aya-Expanse	0.237	0.289	0.440	2.123	2.951	4.143	0.115	0.158	0.148	0.611	0.743	0.813
Qwen3	0.283	0.328	0.438	2.396	3.181	3.954	0.102	0.116	0.097	0.678	0.795	0.825
OLMo-3	0.335	0.382	0.498	2.744	3.801	4.664	**0.084**	**0.100**	0.130	0.726	0.836	0.862
Llama-3.1	0.255	0.289	0.466	2.283	3.003	4.392	0.117	0.160	0.157	0.672	0.762	0.841
Ministral-3	0.234	0.234	0.352	2.203	2.599	3.419	0.113	0.132	0.199	0.653	0.745	0.809
Our injected tokenizers												
Gemma-3 + Greedy (34,809)	0.246	0.223	0.252	1.932	2.152	2.196	0.119	0.174	0.152	0.516	0.566	0.581
Gemma-3 + LIFT-5,000	0.225	0.209	0.248	2.025	2.272	2.474	0.123	0.176	0.164	0.586	0.645	0.718
⊢ + LIFT-10,000	0.227	0.209	0.247	2.006	2.254	2.438	0.122	0.176	0.162	0.576	0.639	0.707
⊢ + LIFT-20,000	0.234	0.214	0.247	**1.967**	**2.213**	**2.357**	0.119	0.173	0.159	**0.548**	**0.613**	**0.671**
Aya-Expanse + Greedy (37,647)	0.301	0.323	0.379	2.142	2.809	3.245	0.065	0.069	0.078	0.533	0.613	0.634
Aya-Expanse + LIFT-5,000	0.238	0.287	0.435	2.099	2.911	4.069	0.110	0.153	0.146	0.601	0.734	0.800
⊢ + LIFT-10,000	0.239	0.283	0.429	2.079	2.856	3.997	0.107	0.148	0.144	0.592	0.724	0.789
⊢ + LIFT-20,000	0.244	0.280	0.415	2.032	2.762	3.828	0.103	0.138	0.136	0.566	0.697	0.756
Qwen3 + Greedy (38,846)	0.326	0.334	0.380	2.268	2.840	3.160	0.052	0.085	0.050	0.559	0.621	0.636
Qwen3 + LIFT-5,000	0.283	0.326	0.433	2.364	3.134	3.880	0.098	0.115	0.096	0.668	0.785	0.813
⊢ + LIFT-10,000	0.284	0.324	0.428	2.329	3.085	3.815	0.093	0.113	0.095	0.652	0.773	0.800
⊢ + LIFT-20,000	0.286	0.319	0.414	2.266	2.986	3.660	0.086	0.112	**0.092**	0.622	0.747	0.769

Short Name	**Model Name**	**Training Stage**	**License**	**Hugging Face Model ID**
Aya 32B	Aya Expanse 32B	Pretrained + post-trained	CC-BY-NC + AUP	`CohereLabs/aya-expanse-32b`
Gemma 27B	Gemma 3 27B PT	Base / pre-trained	Gemma Terms	`google/gemma-3-27b-pt`
gpt-oss 20B	GPT-OSS 20B	Reasoning-oriented	Apache 2.0	`openai/gpt-oss-20b`
Llama 3.1 8B	Llama 3.1 8B	Base / pre-trained	Llama 3.1 Community	`meta-llama/Llama-3.1-8B`
Ministral 14B	Ministral 3 14B Base 2512	Base / pre-trained	Apache 2.0	`mistralai/Ministral-3-14B-Base-2512`
OLMo 32B	OLMo 3 1125 32B	Base / pre-trained	Apache 2.0	`allenai/Olmo-3-1125-32B`
Phi-4	Phi-4	Pretrained + safety post-trained	MIT	`microsoft/phi-4`
Qwen3 32B	Qwen3 32B	Pretraining + post-training	Apache 2.0	`Qwen/Qwen3-32B`

Table 19: Short display names used in the plots, together with the corresponding full model names, training stage, license label, and Hugging Face model identifiers. GPT-4 and GPT-5 refer to tokenizer-only baselines obtained through the `tiktoken` library. License labels are reported according to the corresponding model cards or license files and should be interpreted as concise identifiers rather than substitutes for the full license terms. For readability, the heatmaps and related plots use shortened display names (e.g., *Gemma 27B*, *Qwen3 32B*, *OLMo 32B*) rather than raw repository strings.

Family counts

Top-level family / bucket	Count
Indo-European (*IndoE*)	24
Austronesian (*AustN*)	5
Niger–Congo (*NigerC*)	6
Dravidian (*Drav*)	4
Afro-Asiatic	3
Uralic	2
Turkic	2
Isolate	1
Creole	1
Japonic	1
Koreanic	1
Sino-Tibetan	1
Tai–Kadai	1
Austroasiatic	1
Others	14
Total	53

Script counts

Script / bucket	Count
Latn (*Latn*)	28
Arab (*Arab*)	5
Cyrl (*Cyrl*)	4
Deva	2
Beng	1
Grek	1
Gujr	1
Java	1
Jpan	1
Knda	1
Kore	1
Mlym	1
Mymr	1
Guru	1
Sinh	1
Taml	1
Telu	1
Thai	1
Others	16
Total	53

Resource counts

Resource class	Count
High	17
Mid	13
Low	23
Total	53

Main-text abbreviation	Expanded form
IndoE	Indo-European
AustN	Austronesian
NigerC	Niger–Congo
Drav	Dravidian
Latn	Latin script
Cyrl	Cyrillic script
Arab	Arabic script
Others	all remaining families or scripts not shown individually

Table 20: Composition of the 53-language evaluation set. Left: top-level family counts and resource-class counts. Right: script counts. Bottom: abbreviations used in the grouped family and script summaries in the main text. In the main-text grouped analysis, *Others* aggregates all families outside *IndoE*, *AustN*, *NigerC*, and *Drav*, and all scripts outside *Latn*, *Cyrl*, and *Arab*. These summaries clarify the bracketed group sizes reported in Table 1.

Table 21: Final language catalog used in our experiments, comprising 53 languages. For each language, we report its ISO 639–3 code, ISO 639–1 code (when available), detailed genealogical family path, dominant script (ISO 15924), and resource class. ISO 639–3 and ISO 639–1 codes follow the corresponding ISO standards, and script tags follow ISO 15924. Resource classes are derived following (Joshi et al., 2020) and are aligned with the prior multilingual LLM studies (Singh et al., 2024; Üstün et al., 2024).

Language	ISO 639-3	ISO 639-1	Family (detailed path)	Script	Resources
Arabic	arb	ar	Afro-Asiatic > Semitic > Central Semitic	Arab	High
Basque	eus	eu	Isolate	Latn	High
Bengali	ben	bn	Indo-European > Indo-Iranian > Indo-Aryan	Beng	Mid
Burmese	mya	my	Sino-Tibetan > Lolo-Burmese > Burmese	Mymr	Low
Cebuano	ceb	ceb	Austronesian > Malayo-Polynesian > Philippine	Latn	Low
Chichewa	nya	ny	Niger–Congo > Atlantic–Congo > Benue–Congo > Bantu	Latn	Low
Danish	dan	da	Indo-European > Germanic > North Germanic	Latn	Mid
Dutch	nld	nl	Indo-European > Germanic > West Germanic	Latn	High
Filipino	fil	tl	Austronesian > Malayo-Polynesian > Philippine	Latn	Mid
Finnish	fin	fi	Uralic > Finnic	Latn	Mid
French	fra	fr	Indo-European > Italic > Romance	Latn	High
German	deu	de	Indo-European > Germanic > West Germanic	Latn	High
Greek	ell	el	Indo-European > Hellenic	Grek	Mid
Gujarati	guj	gu	Indo-European > Indo-Iranian > Indo-Aryan	Gujr	Low
Haitian Creole	hat	ht	Creole > French-based	Latn	Low
Hausa	hau	ha	Afro-Asiatic > Chadic > West Chadic	Arab	Low
Hindi	hin	hi	Indo-European > Indo-Iranian > Indo-Aryan	Deva	High
Hungarian	hun	hu	Uralic > Ugric	Latn	High
Igbo	ibo	ig	Niger–Congo > Atlantic–Congo > Volta–Niger	Latn	Low
Indonesian	ind	id	Austronesian > Malayo-Polynesian > Western Malayo-Polynesian	Latn	Mid
Irish	gle	ga	Indo-European > Celtic > Goidelic	Latn	Low
Italian	ita	it	Indo-European > Italic > Romance	Latn	High

Continued on next page

Language	ISO 639-3	ISO 639-1	Family (detailed path)	Script	Resources
Japanese	jpn	ja	Japonic	Jpan	High
Javanese	jav	jv	Austronesian > Malayo-Polynesian > Javanese	Java	Low
Kannada	kan	kn	Dravidian > South Dravidian	Knda	Low
Korean	kor	ko	Koreanic	Kore	Mid
Kyrgyz	kir	ky	Turkic > Common Turkic > Kipchak	Cyrl	Low
Lithuanian	lit	lt	Indo-European > Balto-Slavic > Baltic	Latn	Mid
Malayalam	mal	ml	Dravidian > South Dravidian	Mlym	Low
Marathi	mar	mr	Indo-European > Indo-Iranian > Indo-Aryan	Deva	Low
Persian	fas	fa	Indo-European > Indo-Iranian > Iranian	Arab	High
Polish	pol	pl	Indo-European > Balto-Slavic > Slavic > West Slavic	Latn	High
Portuguese	por	pt	Indo-European > Italic > Romance	Latn	High
Punjabi	pan	pa	Indo-European > Indo-Iranian > Indo-Aryan	Guru	Low
Russian	rus	ru	Indo-European > Balto-Slavic > Slavic > East Slavic	Cyrl	High
Serbian	srp	sr	Indo-European > Balto-Slavic > Slavic > South Slavic	Cyrl	Mid
Shona	sna	sn	Niger–Congo > Atlantic–Congo > Benue–Congo > Bantu	Latn	Low
Sindhi	snd	sd	Indo-European > Indo-Iranian > Indo-Aryan	Arab	Low
Sinhala	sin	si	Indo-European > Indo-Iranian > Indo-Aryan	Sinh	Low
Somali	som	so	Afro-Asiatic > Cushitic > Lowland East Cushitic	Latn	Low
Spanish	spa	es	Indo-European > Italic > Romance	Latn	High
Sundanese	sun	su	Austronesian > Malayo-Polynesian > Sundanese	Latn	Low
Swedish	swe	sv	Indo-European > Germanic > North Germanic	Latn	High
Tamil	tam	ta	Dravidian > South Dravidian	Taml	Mid
Telugu	tel	te	Dravidian > South Central Dravidian	Telu	Low
Thai	tha	th	Tai–Kadai > Kam–Tai > Tai	Thai	Mid
Turkish	tur	tr	Turkic > Oghuz	Latn	High
Ukrainian	ukr	uk	Indo-European > Balto-Slavic > Slavic > East Slavic	Cyrl	Mid
Urdu	urd	ur	Indo-European > Indo-Iranian > Indo-Aryan	Arab	Mid

Continued on next page

Language	ISO 639-3	ISO 639-1	Family (detailed path)	Script	Resources
Vietnamese	vie	vi	Austroasiatic > Vietic	Latn	High
Xhosa	xho	xh	Niger–Congo > Atlantic–Congo > Benue–Congo > Bantu	Latn	Low
Yoruba	yor	yo	Niger–Congo > Atlantic–Congo > Volta–Niger	Latn	Low
Zulu	zul	zu	Niger–Congo > Atlantic–Congo > Benue–Congo > Bantu	Latn	Low