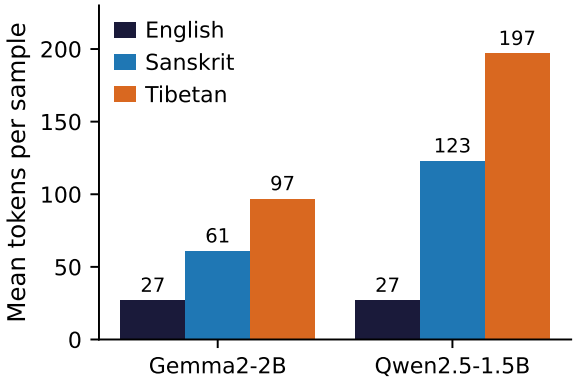
Do We Really Need So Many Tokens? The Performance-Efficiency Trade-off in Tibetan and Sanskrit

Orr Meir Zwebner[1,2], **Kai Golan Hashiloni**[1,2], **Guy Bilitski**[1,2], **Oded Hellman**[1,2], **Nir Marciano**[2], **Erez Yousfan**[1,2], **Shai Fine**[1], **Kfir Bar**[1,2]
[1]Data Science Institute, Reichman University, Herzliya, Israel
[2]Efi Arazi School of Computer Science, Reichman University, Herzliya, Israel
Correspondence: orr.zwebner@post.runi.ac.il

Abstract

Large language models excel in high-resource languages, but their performance on low-resource languages (LRLs) depends mainly on two factors: limited training data and inefficient tokenization. For non-Latin scripts, tokenizers whose vocabularies are largely English produce high token fertility, that is, they split words into more subword tokens per word, increasing sequence length and computational cost. Vocabulary expansion addresses this by adding language-specific tokens, but the trade-offs it introduces are not fully characterized. We evaluate this strategy across architectures and model scales,[1] focusing on two LRLs, Tibetan and Sanskrit. Each new token requires training a new embedding. Our results show that as more tokens are added under limited continued pre-training data, these embeddings remain undertrained. While expansion improves efficiency, performance peaks at small expansion sizes and decreases as the vocabulary grows.



English (12) Some | _festivals | _have | _special | _camping | _areas | _for | _families | _with | _young | _children | .

Sanskrit (31) क | ति | प | य | ▁उत् | स | वे | ▁क्ष | ु | द्र | ▁बाल | क | स्य | ▁परिवार | ं | ▁कृ | ते | ▁विशेष | ▁श | ि | वि | र | ▁क्ष | त्र | ं | ▁इ | ति | ▁भ | व | ति | ▁।

Tibetan (59) ད|ེ་|ས་|ས|ྱི|ོན་|འ|གྲ|འ|་|ཞ|ིག་|ལ|་|ས|་ག|ེ་|ས|ུ|ས|ྲ|ུ|གྱ་|ཡ|ོད་|མ|ཁ|ན་|གི|ི|ཁ|ུ|ི|མ་|ཚ|ང་|ཚ|ེད་|ས|ཀྱ|ར་ག|ོ|ས|་ས|ི་|ས་|གི|ན|་ས|ི|གས་|གི|ས|ལ|་ག|་|ཡ|ོ|ད།།

Figure 1: **Cross-Lingual Tokenization.** *Top:* mean tokens per sample on 1,012 FLORES-IN parallel samples (Gemma2-2B and Qwen2.5-1.5B tokenizers). *Bottom:* Gemma2 tokenization of one aligned sentence across languages; "|" marks subword boundaries. Full sentences and transliterations are provided in Appendix O.

1 Introduction

Language adaptation pre-training (LAPT) is a standard approach to adapt large language models (LLMs) to new languages (Wang et al., 2025; Tejaswi et al., 2024; Mundra et al., 2024; Downey et al., 2023). In LRLs, LAPT aims to bridge two disparities: the *knowledge gap* (Nag et al., 2025; Lu et al., 2025), stemming from the scarcity of target-language text in pre-training, and the *efficiency gap* (Yamaguchi et al., 2026, 2024; Tejaswi et al., 2024; Lin et al., 2025)—the higher number of tokens required to represent text written in LRLs compared to high-resource languages, which increases sequence length and computational cost. While continued pre-training (CPT) addresses the knowledge gap by exposing the model to new data, the efficiency gap requires changes to the vocabulary and embedding matrices (Mundra et al., 2024).

Figure 1 illustrates the efficiency gap on the IndicGenBench (Singh et al., 2024) FLORES-200 dataset: Tibetan and Sanskrit require substantially more tokens than English to represent the same content. This difference highlights how such tokenizers struggle disproportionately with some non-Latin scripts. These tokenizers segment words written in under-represented scripts into long sequences of characters or raw bytes (Hong et al., 2024; Rust et al., 2021; Ahia et al., 2023; Petrov et al., 2023). This over-fragmentation increases inference costs and has been shown to degrade downstream performance (Ali et al., 2024; Rust et al., 2021; Toraman et al., 2023).

Vocabulary expansion is the most common ap-

[1]Code is available at https://github.com/Intellexus-DSI/do-we-really-need-so-many-tokens.

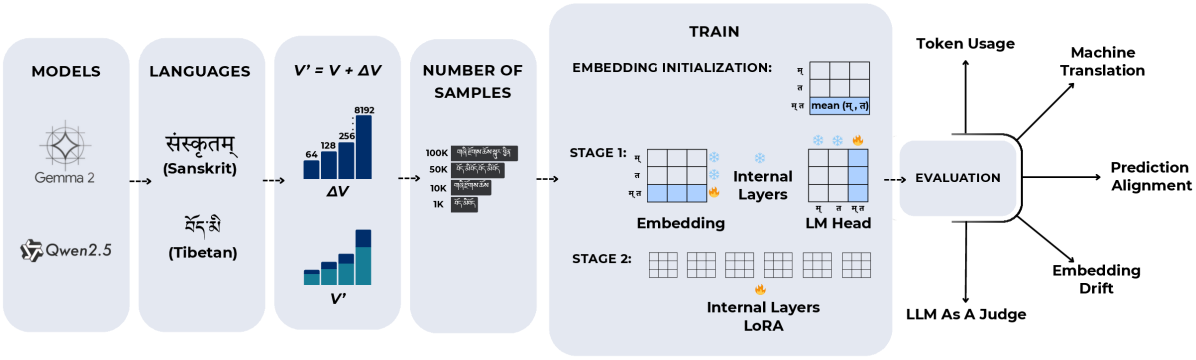


Figure 2: Methodology overview: models, languages, vocabulary expansion, training stages, and evaluation suite.

proach to addressing this gap (Yamaguchi et al., 2026; Hong et al., 2024): it extends the vocabulary $\mathcal{V}$ to $\mathcal{V}'$ and resizes the embedding matrix from $E \in \mathbb{R}^{|\mathcal{V}| \times d}$ to $E' \in \mathbb{R}^{|\mathcal{V}'| \times d}$ (d: hidden dimension). By introducing language-specific tokens, the method shortens sequence length and reduces computational cost. We focus on Tibetan and Sanskrit to examine script-dependent behaviors: Tibetan uses a script substantially under-represented in most open-weight models (Sha et al., 2025), whereas Sanskrit uses Devanagari, a script familiar to most models through high-resource languages like Hindi. As shown in Figure 1, Tibetan requires about 1.5× as many tokens as Sanskrit to represent the same content.

We report token counts (Schmidt et al., 2024) rather than fertility ratios to avoid bias from cross-lingual morphological differences. The challenge is to close both the knowledge and efficiency gaps without harming capabilities such as downstream task performance and generation quality.

Each new token introduces new embeddings that must be learned during continued pre-training, making vocabulary expansion sensitive to the amount of available data (Schmidt et al., 2024; Hewitt, 2021; Mundra et al., 2024; Dobler and de Melo, 2023). To study how this intervention behaves under different data constraints, we systematically vary both the amount of continued pre-training data and the size of the vocabulary expansion. We evaluate this design across multiple model scales and two typologically distinct LRLs, Tibetan and Sanskrit, using machine translation alongside complementary metrics. Figure 2 illustrates the components of our methodology. Our main contributions are as follows:

1. We show that, for Tibetan and Sanskrit, vocabulary expansion exhibits a trade-off between efficiency and performance. Although expanding the vocabulary improves efficiency, the best model performance is obtained with substantially fewer new tokens than typically assumed in high-resource language adaptation (e.g., Cui et al., 2024).

2. We find a script-dependent asymmetry: Sanskrit benefits less from vocabulary expansion despite its complex morphology, because its script is shared with languages already represented in the base vocabulary, unlike Tibetan.

3. We study the impact of vocabulary expansion on model behavior and capabilities using four metrics covering usage and embedding drift of new tokens, in-context token prediction, and generation quality. We apply these metrics to address the limits of relying solely on downstream task scores and the scarcity of generative benchmarks for Tibetan and Sanskrit (Hashiloni et al., 2025).

To guide our investigation, we pose the following research questions:

- **RQ1:** How does the number of new tokens impact the trade-off between efficiency and model performance under varying CPT data constraints?
- **RQ2:** How do these dynamics differ between Sanskrit (written in Devanagari, a well-represented script) and Tibetan (written in an under-represented script)?
- **RQ3:** Do vocabulary expansion effects vary across different model architectures and parameter scales?

2 Related Work

2.1 Vocabulary Expansion

Vocabulary expansion is a known technique for adapting LLMs to new languages, but prior work spans a wide range of expansion sizes. Cui et al. (2024) added 20k tokens when adapting LLaMA to Chinese, training the expanded model with LoRA (Hu et al., 2022). Tejaswi et al. (2024) recommend more moderate expansion (around 10k tokens) for low and mid-resource languages, while Yamaguchi et al. (2026) show that under strict data constraints ($\sim$0.01GB), small expansions ($\leq$1k tokens) are often sufficient. Wang et al. (2025) likewise favor minimal expansion (2k tokens) and report further gains from including parallel data during CPT. For English, however, Schmidt et al. (2024) find no clear relationship between a tokenizer's compression and downstream accuracy.

2.2 Embedding Initialization

The new embeddings must be initialized before training, and prior work differs on how to do this. Hewitt (2021) assigns every new token the mean of all existing embeddings, and Mundra et al. (2024) evaluate that method against more elaborate methods and report it comparable to them. Dobler and de Melo (2023) initialize each new token's embedding as a weighted average of the most similar tokens already in the original vocabulary. Downey et al. (2023) evaluate this method against simpler initialization techniques (mostly script-wise), and find them comparable.

Following prior work (Gee et al., 2022; Liu et al., 2023; Nag et al., 2023; Yamaguchi et al., 2026; Tejaswi et al., 2024; Golan Hashiloni et al., 2025), we initialize the embedding of each new token as the mean of the embeddings of its subword pieces (Section 3).

2.3 Training Strategies

Kim et al. (2024) alternate between freezing and updating embedding layers for Korean, and Sha et al. (2025) adapt LLaMA to Tibetan with a two-stage protocol (embedding alignment followed by LoRA training), expanding the vocabulary by approximately 26k Tibetan tokens. More recently, Yang et al. (2026) extend Qwen2.5-7B for Tibetan with 15k new tokens and scale further to a Mixture-of-Experts architecture. For Sanskrit, Chaudhari et al. (2024) pre-train a generative model from scratch with a custom 12k-token tokenizer designed for Sanskrit text.

2.4 Alternatives to Vocabulary Expansion

Vocabulary expansion is not the only adaptation strategy. Joshi et al. (2025) adapt Nemotron-Mini-4B to Hindi via full CPT on aligned English–Hindi data without adding tokens, while Nag et al. (2025) select both training data and vocabulary tokens for Indic languages, noting that frequency-based selection can favor tokens that are common but not necessarily useful for adaptation.

Across these language adaptation strategies, the joint effect of expansion size and CPT data volume on the efficiency–performance trade-off remains insufficiently characterized, particularly for LRLs and under-represented scripts. Our work evaluates this trade-off directly under controlled conditions across vocabulary sizes, data volumes, model families, and parameter scales for Tibetan and Sanskrit.

3 Methodology

To address our research questions, we evaluate multiple LLMs across varying vocabulary expansion sizes and CPT data volumes in both languages.

3.1 Formalization

Let M be a pre-trained LLM with an existing tokenizer T whose vocabulary is V. The model consists of an input embedding matrix $E_{in} \in \mathbb{R}^{|V| \times d}$ and an output embedding matrix $E_{out} \in \mathbb{R}^{d \times |V|}$ (LM head). These matrices are tied ($E_{in} = E_{out}^T$) (Press and Wolf, 2017) in Gemma2 (Team et al., 2024) (2B, 9B) and Qwen2.5-1.5B (Yang et al., 2025), and un-tied in Qwen2.5-7B.

We define the vocabulary expansion process as the augmentation of V with a set of new tokens ΔV derived from a corpus written in the target language, yielding an expanded vocabulary $V' = V \cup \Delta V$. This requires resizing the input and output embedding matrices to $\mathbb{R}^{(|V|+|\Delta V|) \times d}$ and $\mathbb{R}^{d \times (|V|+|\Delta V|)}$, respectively. The new embeddings must be initialized and trained.

3.2 Experimental Design

Models. We experiment with Gemma2 (2B, 9B) and Qwen2.5 (1.5B, 7B); Qwen2.5-7B is trained on Tibetan only. Gemma2 uses a SentencePiece (BPE)[2] tokenizer (Kudo and Richardson, 2018)

[2]SentencePiece supports BPE and Unigram algorithms.

with a 256k vocabulary; Qwen2.5 uses a BPE tokenizer (Sennrich et al., 2016) with a 151k vocabulary. In all our experiments, vocabulary expansion uses each model's original tokenizer algorithm, since alternative algorithms would require structural modifications to the existing vocabulary that fall outside the scope of this study.

Vocabulary Construction and Expansion. To construct the expanded vocabularies, we run each model's original tokenizer algorithm over a corpus written in the target language. This produces a frequency-ranked list of candidate tokens. We then remove tokens already included in the original model vocabulary and construct expansion sets whose sizes increase on a logarithmic scale: $|\Delta V| \in \{0, 64, 128, \ldots, 8{,}192\}$. Because BPE is deterministic and frequency-based, in both tokenizers the top-k tokens form a superset of the top-j tokens (where $j < k$). Concretely, the expansion set with $|\Delta V| = 128$ is the exact prefix of the set with $|\Delta V| = 256$, containing its first half.

Each expanded vocabulary was constructed from a fixed 100k-sample corpus, independently of CPT data volume (Land and Bartolo, 2024). Once the expanded vocabulary set is obtained, we resize the input and output embedding matrices to accommodate the new tokens, initialize their embeddings, and proceed with CPT as described below in greater detail.

To initialize the embeddings of the new tokens, we adopt a mean-based initialization strategy (Yamaguchi et al., 2026; Tejaswi et al., 2024). Let $T(v) = [t_1, \ldots, t_k]$ denote the subword tokens that compose v under the original vocabulary V. We initialize both the input and output embedding matrices for v as:

$$E'_{\{in,out\}}(v) = \frac{1}{k} \sum_{i=1}^{k} E_{\{in,out\}}(t_i) \quad (1)$$

This approach places new tokens near related tokens in the pre-trained embedding space, providing a meaningful starting point for training.

Model Training. We follow a two-stage training protocol on a mix of monolingual and parallel data. The full dataset details and the sampling protocol are provided in Appendix A; the Tibetan monolingual corpus and its preprocessing are described in Bilitski et al. (2026). We validate this two-stage design in an ablation study (Appendix C).

Stage 1: Embedding Alignment. We freeze all layers and update only the parameters of the new embeddings in E'_{in} and E'_{out}. Training uses the standard Causal Language Modeling (CLM) objective on monolingual data of size $S \in \{1\text{k}, 10\text{k}, 50\text{k}, 100\text{k}, 300\text{k}\}$; the 300k data size is evaluated on a subset of models only.

Stage 2: Continual Pre-Training. We freeze the embeddings and, for computational efficiency, train the model using Low-Rank Adaptation (LoRA) (Hu et al., 2022; Cui et al., 2024) applied to all linear layers. We train on the monolingual data of size S mixed with parallel (English–target) data at a 10% ratio (Lu et al., 2025; Liu et al., 2021). The full set of hyperparameters and computational details is provided in Appendix B.

3.3 Evaluation Framework

We evaluate our models on tasks from the IndicGenBench suite (Singh et al., 2024): machine translation on FLORES-IN (extending the FLORES-200 dataset (NLLB Team et al., 2024)), XorQA-IN, based on Asai et al. (2021), and CrossSum-IN, based on Bhattacharjee et al. (2023). XorQA-IN and CrossSum-IN yielded very low scores for both base and adapted models, consistent with reported IndicGenBench baselines. Consequently, these results are insufficient to draw meaningful conclusions. Further details are provided in Appendix D.

For MT, we use the standard `chrF++` as relative change from each base model, isolating the effect of vocabulary expansion from prior base capability. We study the impact of vocabulary expansion using the following four complementary metrics. Formal definitions, descriptions, and examples appear in Appendix E.

Token usage measures how often new tokens appear in tokenizer outputs on a held-out test set (FLORES-IN). High usage indicates that new tokens are actively selected during segmentation and likely to benefit from training; low usage indicates redundant expansion and rare training updates.

Embedding drift (Biś et al., 2021) measures the cosine similarity (CS) between each new token's initial and trained embeddings. Values near 1 indicate minimal change; lower values reflect substantial updates. Token usage and embedding drift are read jointly: low CS for actively used tokens suggests meaningful integration, whereas low CS combined with low usage may indicate untrained drift rather than meaningful adaptation (Land and Bartolo, 2024) (see Appendix F).

Prediction Alignment comprises two supervised metrics that evaluate next-token prediction against gold FLORES-IN reference sequences. For a sentence x tokenized as $[t_1, \ldots, t_n]$, we compute the model probability $p_\theta(t_i \mid t_{1\ldots i-1})$ assigned to each gold token given its prefix, and take the greedy (argmax, no sampling) prediction.

The first, *recall* over ΔV, measures accuracy on tokens from the newly added vocabulary. The second, *specificity* over V, measures accuracy on original-vocabulary tokens, detecting regression after vocabulary expansion. These metrics quantify adoption of the new vocabulary alongside preservation of original language modeling performance.

LLM-as-a-Judge provides a qualitative measure of open-ended generation quality. We select 50 Tibetan sentence prefixes from FLORES-IN, generate continuations from each evaluated model, score them using Claude-4.5-Sonnet on a five-point scale (Very bad=1 to Very good=5) covering coherence, grammaticality, and relevance, and report the average score per model. Due to computational constraints, this analysis covers only the Tibetan models trained with 100k samples and the corresponding base model.

To assess judge reliability, we collect ratings from six Tibetan language experts on a subset of 50 outputs generated by a single model configuration (Gemma2-2B, 128 tokens, 100k samples) and compute the correlation between human and LLM scores, following Bavaresco et al. (2025). Full prompts, instructions, and evaluation details are provided in Appendix G.

4 Results

4.1 RQ1: Vocabulary Expansion vs. Performance under Data Constraints

Efficiency. Figure 3 shows the mean number of tokens per FLORES-200 sample. For Gemma2-2B on Tibetan, the mean token count at the largest expansion is reduced by 2.5×. However, these gains scale non-linearly: the first 64 new tokens reduce the mean from 97 to 84 (∼0.21 tokens saved per new token)[3]; at $|\Delta V| = 1{,}024$ the mean reaches 54 (∼0.042 tokens saved per new token), and at $|\Delta V| = 8{,}192$ it drops to 39 (∼0.007 tokens saved per new token). This curve exhibits clear diminishing returns: larger vocabularies yield progressively smaller efficiency gains.

[3] $(\bar{t}_0 - \bar{t}_k)/k$. $\bar{t}_k$: mean tokens per sample at $|\Delta V| = k$.

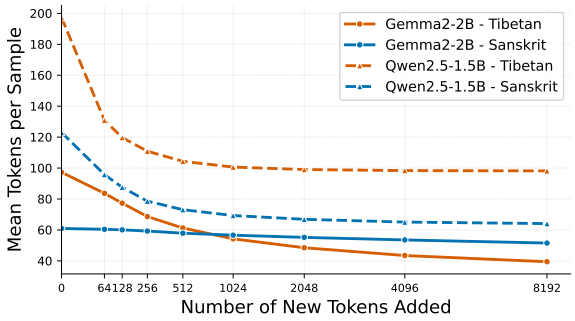


Figure 3: Impact of vocabulary expansion on average number of tokens over the FLORES dataset.

Its knee point (Satopaa et al., 2011)—the expansion size beyond which further tokens yield small efficiency gain—falls at $|\Delta V| = 2{,}048$ for Gemma2-2B on Tibetan, which lets us bound the expansion sizes worth considering before any training. The bound depends only on the tokenizer and a sample of target-language text, and it is an upper bound rather than a recommended size.

Machine Translation Performance. Figure 4 shows relative chrF++ improvements over the base model. Even without expansion ($|\Delta V| = 0$), CPT on 1k samples yields a +187.2% gain, highlighting its strong effect in extremely low-resource settings. Performance peaks at 64–128 new tokens, while larger expansions consistently yield lower results. The best scores are concentrated in higher data regimes (typically 100k), and 8,192 tokens produce the lowest scores in most data regimes, suggesting that large expansions can cause performance degradation under limited data (Table 7, Appendix N).

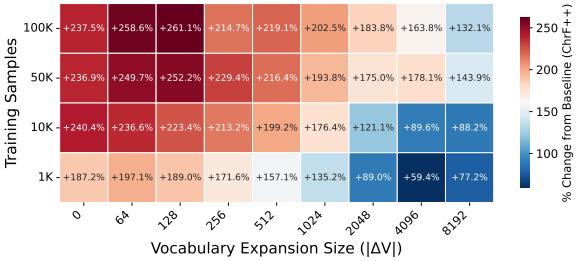


Figure 4: Gemma2-2B: MT Performance, English → Tibetan, % improvement over base (chrF++ 5.17).

Prediction Alignment (Recall of ΔV). Recall over ΔV for Gemma2-2B on Tibetan (Figure 5) depends on both training size S and expansion size $|\Delta V|$. Recall is negligible at 1k samples ($< 5\%$ across all $|\Delta V|$) and rises sharply with S, reaching $\approx 23\%$ at 50k, with modest further gains at 100k. Recall also declines as $|\Delta V|$ grows:

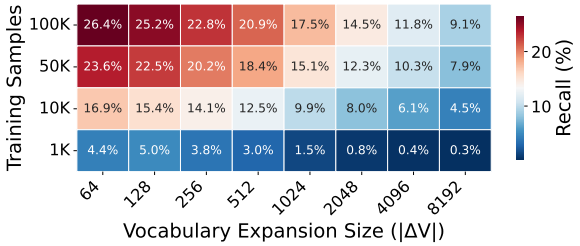


Figure 5: Gemma2-2B on Tibetan: Recall of ΔV (fraction of ΔV tokens correctly predicted).

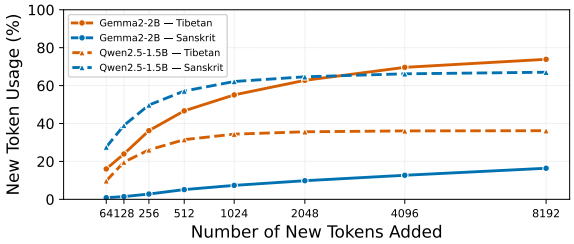


Figure 6: Percentage of new tokens in tokenized FLORES-IN text.

with 8,192 new tokens and 100k samples it reaches 9%, lower than the 10% obtained with 1,024 tokens and only 10k samples, consistent with each new token appearing in training less often as $|\Delta V|$ grows under a fixed amount of data.

Generation Quality (LLM-as-a-Judge). As shown in Table 1, all adapted models outperform the base model, but generation quality declines with vocabulary size, with large expansions yielding the lowest scores. Unlike MT and Recall, the highest result is obtained with no new tokens ($|\Delta V| = 0$, +76.0%); see Appendix N for further discussion.

$\lvert\Delta V\rvert$	**0**	**64**	**128**	**256**	**512**	**1024**	**2048**	**4096**	**8192**
Imp.	**+76%**	+54%	+52%	+34%	+28%	+15%	+12%	+14%	+11%

Table 1: LLM-as-a-Judge relative improvement for Gemma2-2B on Tibetan (100k samples). Base model score: 1.3 out of 5.

Summary (RQ1). Under our setup, a clear efficiency–performance trade-off emerges. Although efficiency improves with vocabulary size, most of the gain is realized at small expansions, with further increases beyond 1,024 tokens yielding progressively smaller returns. In contrast, MT and recall peak at small expansions (64–128 tokens) and decline as more tokens are added, while generation quality is highest without expansion and decreases with vocabulary size.

Although more data improves alignment, it does not prevent degradation at larger sizes: recall over ΔV declines as $|\Delta V|$ grows at a fixed amount of data (Figure 5), suggesting that the new embeddings remain under-trained as more tokens are added. Together, these trends indicate that small vocabulary expansions offer the best efficiency-performance balance under limited CPT data.

4.2 RQ2: Script Representation

Token Usage. We observe clear differences between Sanskrit, written in Devanagari, a well-represented script, and Tibetan, written in an under-represented script. On FLORES-200, the usage of new tokens (Figure 6) in Tibetan increases sharply with vocabulary size—from 16% at $|\Delta V| = 64$ to 74% at $|\Delta V| = 8{,}192$. In contrast, Sanskrit shows consistently low usage, reaching only 16.4% even at the largest expansion. This suggests that, because Devanagari tokens already exist in the base vocabulary, many new Sanskrit tokens are unnecessary and redundant.

Machine Translation Performance. Figure 7 presents the relative performance for Sanskrit, whose Devanagari script is already covered in the base vocabulary. Compared to Tibetan, two notable differences emerge. Base capabilities differ substantially: the Gemma2-2B base model reaches 15.61 chrF++ on Sanskrit but only 5.17 on Tibetan. Yet gains from CPT are modest: Sanskrit peaks at 21.7% (50k samples, 64 tokens), compared to over 200% for Tibetan. With only 1k samples, Sanskrit performance even drops below the base model, showing that token expansion and CPT do not necessarily yield improvements. Regarding vocabulary size, Sanskrit shows no clear trend like the 64–128 peak observed for Tibetan. Results are more scattered, with some decline at large expansions (4,096–8,192 tokens) in higher data regimes.

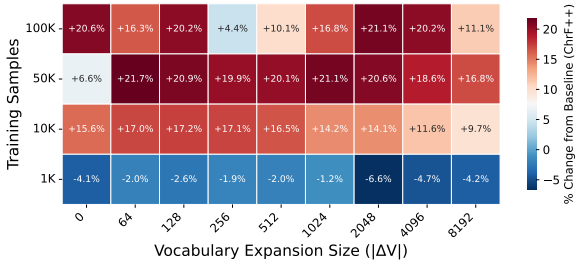


Figure 7: Gemma2-2B: MT performance, English → Sanskrit, % improvement over base (chrF++ 15.61).

Embedding Drift. Figure 8 examines the behavior of new tokens by measuring cosine similarity (CS) between their initialized and final embeddings,

using the Gemma2-2B model trained on 100k samples for both languages. For Tibetan, smaller expansions lead to stronger adaptation. With 64 tokens, median CS is lowest (0.56), indicating substantial updates; combined with the high new-token usage observed for Tibetan (Figure 6), this reflects meaningful integration rather than untrained drift (Land and Bartolo, 2024; Yu et al., 2022; Biś et al., 2021). As vocabulary expansion increases to 8,192 tokens, median CS rises to 0.78 (with most values between 0.73 and 0.83), meaning each token receives weaker updates as updates are distributed across more new embeddings.

Sanskrit shows a different pattern. At $|\Delta V| = 64$, median CS is very high (0.913), meaning over half the tokens barely move from initialization—consistent with their low usage and indicating that these tokens receive few gradient updates during training. Across all expansion sizes, the upper quartile remains near 1.0, indicating that a substantial fraction of new Sanskrit tokens stays close to initialization regardless of $|\Delta V|$. At the same time, in larger expansions (2,048–8,192), the lower quartile ranges from 0.76 to 0.84, meaning that hundreds of tokens drift substantially more than the median token at $|\Delta V| = 64$ (0.913). This points to a potential limitation of frequency-based selection of new Sanskrit tokens: the most frequent ones dominate small expansions, while tokens that drift more during training are ranked lower by frequency and only enter at larger expansion sizes. A direct measurement of this inefficiency: at $|\Delta V| = 8{,}192$ with 100k samples, 48.6% of the new Sanskrit tokens never appear in the tokenized training set, compared to 9.4% for Tibetan (Table 5(a)). Figure 18 in Appendix H confirms this pattern visually: tokens that drift substantially appear at all frequency ranks, not only among the most frequent.

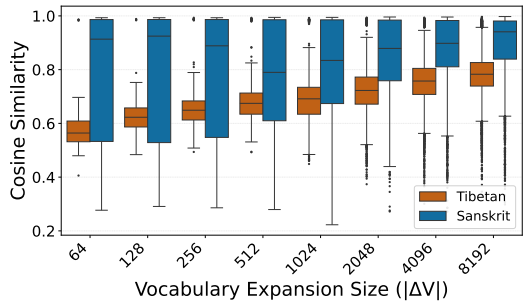


Figure 8: Gemma2-2B: Embedding drift (CS) of new tokens after training on 100k samples.

Prediction Alignment. As observed in Figure 19, Appendix I, recall further highlights Sanskrit's distinct behavior. Unlike the clear monotonic trends observed in Tibetan, Sanskrit exhibits no consistent pattern: recall at 64 and 8,192 tokens is comparable. Although a local peak appears at 512 tokens, it does not yield higher MT gains (10.1%), suggesting it is not a meaningful trend. Specificity analysis (Figure 20, Appendix I) highlights a clear contrast. In Tibetan, specificity decreases as vocabulary expands (18.8% to 8.2%), indicating interference with the original vocabulary. In Sanskrit, specificity remains stable (≈12–16%), suggesting that new tokens do not interfere with predictions on the original vocabulary.

Summary (RQ2). We conclude that the impact of vocabulary expansion differs between Tibetan and Sanskrit, driven primarily by how well their scripts are covered in the base vocabulary. For Tibetan, the new tokens are heavily used and adapt strongly at small expansions, but this adaptation weakens as the vocabulary grows (Figure 8). For Sanskrit, vocabulary expansion has limited impact across our metrics. Across these metrics, the most frequent new Sanskrit tokens are not necessarily the most useful ones to add to the vocabulary, pointing to a potential limitation of using frequency-based algorithms such as BPE for vocabulary expansion.

4.3 RQ3: Impact of Architecture and Scale

We compare two model families across scales—Gemma2 (2B, 9B) and Qwen2.5 (1.5B, 7B)—to examine how vocabulary expansion behaves under different architectures and parameter counts.

4.3.1 Architecture

Sequence Length and Token Usage. Qwen2.5-1.5B shows a clear efficiency ceiling not observed in Gemma2-2B. For Tibetan (Figure 3), the mean token count drops from 197 to 100 at $|\Delta V| = 1{,}024$, but barely decreases further (98 at 8,192), saturating around 1,024 new tokens. Gemma2-2B, starting from a much lower 97 mean tokens, shows a more gradual decline without a clear plateau. For Sanskrit, Qwen2.5-1.5B exhibits a similar plateau, while Gemma2-2B's token count remains consistently low across $|\Delta V|$. The plateau is mirrored in the usage of new tokens (Figure 6): despite Qwen2.5-1.5B's much higher baseline fragmentation for Tibetan (197 vs. 97 mean tokens), it uses fewer new Tibetan tokens than Gemma2-2B across

all expansion sizes. Within Qwen2.5-1.5B, new-token usage is higher for Sanskrit than for Tibetan, even though Tibetan's baseline fragmentation is substantially higher. Together, these patterns show that baseline fragmentation alone does not predict new-token usage.

Machine Translation and Prediction Alignment. We provide a global comparison of MT performance across the smaller models, languages, and expansion sizes in Figure 22, Appendix J. Qwen2.5-1.5B exhibits balanced base capabilities across Sanskrit (11.48) and Tibetan (10.60), in contrast to the marked disparity observed in Gemma2-2B (15.61 vs. 5.17, respectively). For Tibetan, Qwen2.5-1.5B peaks around $|\Delta V| = 512$ in the higher data regimes (50k–100k) and does not show a sharp decline for $|\Delta V| \geq 1{,}024$. Gemma2-2B, in contrast, peaks earlier at $|\Delta V| = 64$–128 and degrades more clearly as the expansion grows. This is consistent with the sequence-length plateau described above: in Qwen2.5-1.5B, both sequence length and MT performance stop changing meaningfully around $|\Delta V| = 1{,}024$. For Sanskrit, Qwen2.5-1.5B shows a clear peak at small expansions ($|\Delta V| \leq 256$) followed by substantial degradation at larger sizes, while Gemma2-2B shows scattered gains with no clear peak or degradation pattern. Across all four model-language pairs, MT performance has a positive correlation with the number of CPT samples available per new token (Table 7, Appendix N).

Recall (Figure 24, Appendix K) is consistently higher in Qwen2.5-1.5B than in Gemma2-2B. However, this advantage does not translate proportionally into downstream MT performance, indicating that stronger token-level prediction alone does not guarantee better MT.

Embedding Drift. Embedding drift in Qwen2.5-1.5B (Figure 9) shows low cosine similarity across both scripts (median ≤ 0.28), decreasing with expansion size. At first glance, this suggests strong adaptation. We test this interpretation by comparing new tokens that appear at least once as a target during training (*seen*) with those that never do (*unseen*). For Gemma2-2B, seen tokens drift substantially more than unseen ones (median CS gap $+0.20$ for Tibetan, $+0.14$ for Sanskrit, Table 5(a)). For Qwen2.5-1.5B, the gap is $+0.03$ and -0.03: unseen tokens drift as much as seen ones, so CS here does not reliably reflect adaptation.

As a complementary measurement, we compare the L2 norm of each new embedding before and after training, using norm as an indicator of under-trained tokens (Land and Bartolo, 2024). In both models, seen tokens grow substantially in norm while unseen tokens stay close to their initialization (Table 5(b), Appendix F). Additional analysis (Appendix F) shows that the new embeddings in Qwen2.5-1.5B cluster in a narrow region of the embedding space, consistent with the representation degeneration problem (Gao et al., 2019; Biś et al., 2021; Yu et al., 2022).

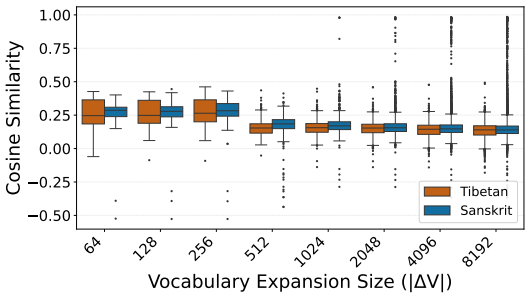


Figure 9: Qwen2.5-1.5B: Embedding drift (CS) of new tokens after training on 100k samples.

These differences may stem from multiple factors distinguishing the two model families, including base vocabulary and its coverage of the target languages, pre-training data, tokenizer algorithm, and other architectural and training choices.

4.3.2 Scale

Gemma2-9B. For Tibetan (Figure 10), MT declines for $|\Delta V| \geq 1{,}024$, with best results typically at 64 tokens in the 50k–300k regimes. In Sanskrit (Figure 26, Appendix L), relative gains are larger (up to ≈70%), likely reflecting the lower Sanskrit baseline of Gemma2-9B (7.56) compared to Gemma2-2B (15.61) rather than a stronger expansion effect. The distribution of Sanskrit results mirrors the smaller model: 1k samples yield minimal or negative gains, no clear degradation pattern emerges in the larger expansion sizes ($|\Delta V| \geq 1{,}024$), and the lowest scores in higher data regimes tend to occur at $|\Delta V| = 8{,}192$. Unlike Tibetan, increasing the training data to 300k does not improve Sanskrit performance for most expansion sizes.

Embedding drift analysis (Figure 25) shows that Gemma2-9B has higher cosine similarity than the 2B model, meaning weaker embedding adaptation at the same data volume. Reaching the median CS observed in Gemma2-2B at 100k requires 300k samples for Gemma2-9B.

Qwen2.5-7B. Due to resource limits, Qwen2.5-7B was evaluated only on Tibetan (Figure 21, Ap-

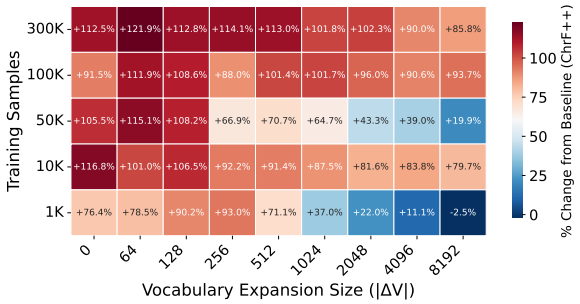


Figure 10: Gemma2-9B: MT Performance, English → Tibetan, % improvement over base (chrF++ 7.67).

pendix M). With a strong baseline (18.08), MT performance remains largely uniform across all expansion volumes. Modest gains (up to $+10.5\%$) appear only at the 50k–100k samples; smaller volumes fall below baseline. The only consistent trend across $|\Delta V|$ is that $|\Delta V| \in \{0, 64\}$ yields the lowest scores at 50k–100k—the opposite of what we observe in the other models, where small expansions are typically the safest choice. Given these limited variations across expansion sizes, larger $|\Delta V|$ values may be worth considering when efficiency gains (§ 4.1) outweigh the small performance differences.

Summary (RQ3). Gemma2-2B and Qwen2.5-1.5B follow largely similar trends across our evaluation suite, with the strongest performance at smaller expansions rather than the larger ones. They differ mainly in efficiency, adaptation to Sanskrit, and embedding drift. Qwen2.5-1.5B reaches an efficiency ceiling around 1,024 new tokens, beyond which sequence length barely decreases. Gemma2-2B, in contrast, continues to shorten sequences as tokens are added, though with the diminishing per-token returns reported in § 4.1. We also observe that in Qwen2.5-1.5B, new embeddings drift heavily regardless of whether they appear in training (Appendix F), limiting the reliability of cosine similarity as an adaptation indicator in this architecture.

At larger scales, these trends do not carry over uniformly: Gemma2-9B mainly benefits from small expansions ($|\Delta V| \approx 64$) but requires more training data to reach comparable embedding adaptation, while Qwen2.5-7B exhibits largely uniform performance across expansion volumes, with modest improvements only in higher data regimes. Across all models and scales, gains from vocabulary expansion appear most consistently at higher data volumes (50k–100k).

5 Conclusion

We examine vocabulary expansion for Tibetan and Sanskrit across model architectures and scales. CPT alone already yields large gains under extreme data scarcity (+187% MT with only 1k Tibetan samples), with the largest relative gains where base capabilities are lowest. Across MT and our four evaluation metrics, performance typically peaks at small vocabulary expansions (around 64 new tokens) and decreases as the vocabulary grows (Figure 23, Appendix N), while efficiency continues to improve at diminishing rates. In our measurements, the degradation at larger expansions points primarily to the new embeddings remaining under-trained. For Tibetan, new-token usage reaches up to 74%, reflecting strong reliance on new tokens. For Sanskrit, usage stays below 16.4%, indicating that the base vocabulary already covers most Devanagari subwords. Overall, for Tibetan and Sanskrit, vocabulary expansion improves efficiency, but the strongest performance gains come from small expansion sizes and higher data volumes across most architectures and scales, with the specific configuration depending on the script and model.

Future work includes extending the analysis to additional low-resource languages, developing token selection algorithms that are aware of the base vocabulary and comparing them with frequency-based methods such as BPE, and investigating the effect of vocabulary expansion on the embedding space and its geometry across architectures.

Acknowledgments

We thank Dorji Wangchuk, Orna Almogi, Nicola Bajetta, Sonam Choden, Sonam Jamtsho, Andrea Cornelius, Rebecca Sundén, Jingyi Yang, Harunaga Isaacson, Ryan Conlon, Shanshan Jia, and Yadi Qu for their contributions to this research, including the expert ratings used to validate the LLM-as-a-Judge scores (Appendix G), as part of their ongoing work on the Intellexus project. We thank Chen Yoeli Zwebner for the design of Figure 2. This study is supported in part by the European Research Council (Intellexus, Project No. 101118558). Views and opinions expressed are, however, those of the authors only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authorities can be held responsible for them.

Limitations

Our study, although providing valuable insights into vocabulary expansion for low-resource languages, is subject to several limitations. First, our research is restricted to two languages, Sanskrit and Tibetan, and a specific set of model architectures and scales due to limited computational resources and data availability. While the paper refers to low-resource languages and under-represented scripts more broadly, our empirical findings concern Tibetan and Sanskrit specifically, and do not necessarily extend to other low-resource languages, nor to medium- or high-resource languages written in non-Latin scripts. Establishing that small expansions are generally preferable for low-resource language adaptation would require broader research and is left for future work.

Second, the Sanskrit and Tibetan corpora differ in source composition and content type, which may introduce variability in training outcomes. We report training volume in samples rather than tokens, since each sample is a separate semantic unit while per-sample token counts may differ across languages and tokenizers. The data sources for each language are described in Appendix A.

Third, the high computational cost of extensive experimentation necessitated using a single random seed, a single embedding initialization strategy (mean-based, drawing on prior work in language adaptation), and specific hyperparameter configurations (such as training stages and LoRA versus Full CPT); we kept these fixed across all expansion sizes, data volumes and models rather than comparing alternatives, so that the comparison isolates the two factors we vary; different settings might alter the observed results.

Fourth, comparisons between the Qwen2.5 and Gemma2 model families are subject to multiple confounds beyond the choice of tokenizer algorithm (BPE for Qwen2.5, SentencePiece (BPE) for Gemma2), including differences in base vocabulary size (151k vs. 256k), in the composition and scale of pre-training data (which is not fully publicly disclosed), and in whether input and output embedding matrices are tied: Gemma2-2B/9B and Qwen2.5-1.5B use tied embeddings, while Qwen2.5-7B uses untied matrices. This last factor may affect how new embeddings adapt during training, but we do not isolate its contribution from the others listed above.

Finally, our conclusions focus on a single downstream task (machine translation) and rely primarily on the FLORES-IN test set, which served as the basis for four out of the five evaluation metrics reported.

Ethics Statement

We use publicly available datasets and models in accordance with their licenses, as detailed in Appendix P. No personally identifiable information was processed in the course of dataset construction, which assembles open-source textual material and therefore did not require ethics review board approval.

For the LLM-as-a-Judge validation (Appendix G), we collected ratings from six Tibetan language experts collaborating on the broader research project of which this work is a part. They are scholars whose research focuses on Tibetan texts and are fluent in Tibetan; they contributed to this evaluation as part of their ongoing project work rather than as recruited or paid annotators. All experts agreed to participate with informed consent, were aware of how their ratings would be used, and were not informed of the LLM judge scores during evaluation. This evaluation was not subject to ethics review board approval, as the raters participated as co-researchers on the broader project rather than as human subjects recruited for this study. Because the task was part of their ongoing project work, no per-task payment was provided for this specific evaluation. The full instructions provided to the experts are shown in Figure 16; the task involved no exposure to offensive content or collection of personal information. Our work is intended solely for research purposes, and we believe it poses no foreseeable risks.

References

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

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, and 2 others. 2024. Tokenizer choice for LLM training: Negligible or crucial? In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 3907–3924, Mexico City, Mexico. Association for Computational Linguistics.

Akari Asai, Jungo Kasai, Jonathan Clark, Kenton Lee, Eunsol Choi, and Hannaneh Hajishirzi. 2021. XOR QA: Cross-lingual open-retrieval question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 547–564, Online. Association for Computational Linguistics.

Anna Bavaresco, Raffaella Bernardi, Leonardo Bertolazzi, Desmond Elliott, Raquel Fernández, Albert Gatt, Esam Ghaleb, Mario Giulianelli, Michael Hanna, Alexander Koller, Andre Martins, Philipp Mondorf, Vera Neplenbroek, Sandro Pezzelle, Barbara Plank, David Schlangen, Alessandro Suglia, Aditya K Surikuchi, Ece Takmaz, and Alberto Testoni. 2025. LLMs instead of human judges? a large scale empirical study across 20 NLP evaluation tasks. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 238–255, Vienna, Austria. Association for Computational Linguistics.

Abhik Bhattacharjee, Tahmid Hasan, Wasi Uddin Ahmad, Yuan-Fang Li, Yong-Bin Kang, and Rifat Shahriyar. 2023. CrossSum: Beyond English-centric cross-lingual summarization for 1,500+ language pairs. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2541–2564, Toronto, Canada. Association for Computational Linguistics.

Guy Bilitski, Lev Shechter, Sonam Jamtsho, Nir Marciano, Nicola Bajetta, Rebecca Sunden, Omri Drori, Kai Golan Hashiloni, Orr Zwebner, Asaf Shina, Orna Almogi, Dorji Wangchuk, and Kfir Bar. 2026. Automatic segmentation of Classical Tibetan texts into autochthonous and allochthonous regions. In *Proceedings of the Fifteenth Language Resources and Evaluation Conference*, pages 1017–1030, Palma de Mallorca, Spain. ELRA Language Resource Association.

Daniel Biś, Maksim Podkorytov, and Xiuwen Liu. 2021. Too much in common: Shifting of embeddings in transformer language models and its implications. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5117–5130, Online. Association for Computational Linguistics.

Rhugved Pankaj Chaudhari, Bhakti Jadhav, Pushpak Bhattacharyya, and Malhar Kulkarni. 2024. SansGPT: Advancing generative pre-training in Sanskrit. In *Proceedings of the 21st International Conference on Natural Language Processing (ICON)*, pages 432–441, AU-KBC Research Centre, Chennai, India. NLP Association of India (NLPAI).

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Yiming Cui, Ziqing Yang, and Xin Yao. 2024. Efficient and effective text encoding for Chinese LLaMA and Alpaca. *Preprint*, arXiv:2304.08177.

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

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

Jun Gao, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tie-Yan Liu. 2019. Representation degeneration problem in training natural language generation models. In *International Conference on Learning Representations*.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, and 5 others. 2024. The language model evaluation harness. Zenodo, `https://doi.org/10.5281/zenodo.5371628`. Source code: `https://github.com/EleutherAI/lm-evaluation-harness`.

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

Kai Golan Hashiloni, Brenda Kasabe Nokai, Michal Shevach, Esthy Shemesh, Ronit Bartin, Anna Bergrin, Liran Harel, Nachum Dershowitz, Liat Nadai Arad, and Kfir Bar. 2025. Building patient journeys in Hebrew: A language model for clinical timeline extraction. In *Proceedings of the Workshop on Large Language Models and Generative AI for Health Informatics (IJCAI 2025)*, Montreal, Canada.

Kai Golan Hashiloni, Shay Cohen, Asaf Shina, Jingyi Yang, Orr Meir Zwebner, Nicola Bajetta, Guy Biliitski, Rebecca Sundén, Guy Maduel, Ryan Conlon,

Ari Barzilai, Daniel Mass, Shanshan Jia, Aviv Naaman, Sonam Choden, Sonam Jamtsho, Yadi Qu, Harunaga Isaacson, Dorji Wangchuk, and 3 others. 2025. DharmaBench: Evaluating language models on buddhist texts in Sanskrit and Tibetan. In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pages 2088–2110, Mumbai, India. The Asian Federation of Natural Language Processing and The Association for Computational Linguistics.

John Hewitt. 2021. Initializing new word embeddings for pretrained language models. `https://www.cs.columbia.edu/~johnhew/vocab-expansion.html`. Accessed 2026-09-03.

Jimin Hong, Gibbeum Lee, and Jaewoong Cho. 2024. Accelerating multilingual language model for excessively tokenized languages. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 11095–11111, Bangkok, Thailand. Association for Computational Linguistics.

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

Raviraj Joshi, Kanishk Singla, Anusha Kamath, Raunak Kalani, Rakesh Paul, Utkarsh Vaidya, Sanjay Singh Chauhan, Niranjan Wartikar, and Eileen Long. 2025. Adapting multilingual LLMs to low-resource languages using continued pre-training and synthetic corpus: A case study for Hindi LLMs. In *Proceedings of the First Workshop on Natural Language Processing for Indo-Aryan and Dravidian Languages*, pages 50–57, Abu Dhabi. Association for Computational Linguistics.

Seungduk Kim, Seungtaek Choi, and Myeongho Jeong. 2024. Efficient and effective vocabulary expansion towards multilingual large language models. *Preprint*, arXiv:2402.14714.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

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

Nankai Lin, Peijian Zeng, Weixiong Zheng, Shengyi Jiang, Dong Zhou, and Aimin Yang. 2025. Rethinking vocabulary augmentation: Addressing the challenges of low-resource languages in multilingual models. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 2919–2934, Abu Dhabi, UAE. Association for Computational Linguistics.

Siyang Liu, Naihao Deng, Sahand Sabour, Yilin Jia, Minlie Huang, and Rada Mihalcea. 2023. Task-adaptive tokenization: Enhancing long-form text generation efficacy in mental health and beyond. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 15264–15281, Singapore. Association for Computational Linguistics.

Zihan Liu, Genta Indra Winata, and Pascale Fung. 2021. Continual mixed-language pre-training for extremely low-resource neural machine translation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2706–2718, Online. Association for Computational Linguistics.

Kaiwen Lu, Yating Yang, Fengyi Yang, Rui Dong, Bo Ma, Aihetamujiang Aihemaiti, Abibilla Atawulla, Lei Wang, and Xi Zhou. 2025. Low-resource language expansion and translation capacity enhancement for LLM: A study on the Uyghur. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 8360–8373, Abu Dhabi, UAE. Association for Computational Linguistics.

Ayush Maheshwari, Ashim Gupta, Amrith Krishna, Atul Kumar Singh, Ganesh Ramakrishnan, Anil Kumar Gourishetty, and Jitin Singla. 2024. Samayik: A benchmark and dataset for English-Sanskrit translation. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 14298–14304, Torino, Italia. ELRA and ICCL.

Jacob Moore. 2024. Tibetan to English translation dataset. HuggingFace dataset repository, `https://huggingface.co/datasets/billingsmoore/tibetan-to-english-translation-dataset`. Scraped from Lotsawa House; CC BY-NC 4.0; accessed 2026-09-06.

Nandini Mundra, Aditya Nanda Kishore Khandavally, Raj Dabre, Ratish Puduppully, Anoop Kunchukuttan, and Mitesh M Khapra. 2024. An empirical comparison of vocabulary expansion and initialization approaches for language models. In *Proceedings of the 28th Conference on Computational Natural Language Learning*, pages 84–104, Miami, FL, USA. Association for Computational Linguistics.

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

Arijit Nag, Bidisha Samanta, Animesh Mukherjee, Niloy Ganguly, and Soumen Chakrabarti. 2023. Entropy-guided vocabulary augmentation of multilingual language models for low-resource tasks. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8619–8629, Toronto, Canada. Association for Computational Linguistics.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, and 20 others. 2024. Scaling neural machine translation to 200 languages. *Nature*, 630(8018):841–846.

Aleksandar Petrov, Emanuele La Malfa, Philip H. S. Torr, and Adel Bibi. 2023. Language model tokenizers introduce unfairness between languages. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

Sonam Phuntsog. 2024. Tibetan monolingual corpus. HuggingFace dataset repository, `https://huggingface.co/datasets/spsither/tibetan_monolingual_A_filtered_deduped`. No license declared at the repository; accessed 2026-08-31.

Ofir Press and Lior Wolf. 2017. Using the output embedding to improve language models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 157–163, Valencia, Spain. Association for Computational Linguistics.

Phillip Rust, Jonas Pfeiffer, Ivan Vulić, Sebastian Ruder, and Iryna Gurevych. 2021. How good is your tokenizer? on the monolingual performance of multilingual language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3118–3135, Online. Association for Computational Linguistics.

Ville Satopaa, Jeannie Albrecht, David Irwin, and Barath Raghavan. 2011. Finding a "Kneedle" in a haystack: Detecting knee points in system behavior. In *2011 31st International Conference on Distributed Computing Systems Workshops*, pages 166–171.

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

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Jiu Sha, Mengxiao Zhu, Chong Feng, and Jizhuoma Ci. 2025. Tibetan-LLaMA 2: Large language model for Tibetan. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 24(12):141:1–141:19.

Harman Singh, Nitish Gupta, Shikhar Bharadwaj, Dinesh Tewari, and Partha Talukdar. 2024. IndicGenBench: A multilingual benchmark to evaluate generation capabilities of LLMs on Indic languages. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11047–11073, Bangkok, Thailand. Association for Computational Linguistics.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, and 179 others. 2024. Gemma 2: Improving open language models at a practical size. *Preprint*, arXiv:2408.00118.

Atula Tejaswi, Nilesh Gupta, and Eunsol Choi. 2024. Exploring design choices for building language-specific LLMs. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 10485–10500, Miami, Florida, USA. Association for Computational Linguistics.

Cagri Toraman, Eyup Halit Yilmaz, Furkan Şahinuç, and Oguzhan Ozcelik. 2023. Impact of tokenization on language models: An analysis for Turkish. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 22(4):116:1–116:21.

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

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. 2020. CCNet: Extracting high quality monolingual datasets from web crawl data. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4003–4012, Marseille, France. European Language Resources Association.

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

Atsuki Yamaguchi, Aline Villavicencio, and Nikolaos Aletras. 2026. How can we effectively expand the vocabulary of LLMs with 0.01GB of target language text? *Computational Linguistics*, 52(1):295–330.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, and 23 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Lei Yang, Leiyu Pan, Bojian Xiong, Renren Jin, Shaowei Zhang, Yue Chen, Ling Shi, Jiang Zhou, Junru Wu, Zhen Wang, Jianxiang Peng, Juesi Xiao, Tianyu Dong, Zhuowen Han, Zhuo Chen, Yuqi Ren, and Deyi Xiong. 2026. From curated data to scalable models: Continual pre-training of dense and MoE large language models for Tibetan. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 40167–40182, San Diego, California, United States. Association for Computational Linguistics.

Sangwon Yu, Jongyoon Song, Heeseung Kim, Seongmin Lee, Woo-Jong Ryu, and Sungroh Yoon. 2022. Rare tokens degenerate all tokens: Improving neural text generation via adaptive gradient gating for rare token embeddings. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 29–45, Dublin, Ireland. Association for Computational Linguistics.

Chen Zhang, Mingxu Tao, Quzhe Huang, Jiuheng Lin, Zhibin Chen, and Yansong Feng. 2024. MC^2: Towards transparent and culturally-aware NLP for minority languages in China. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8832–8850, Bangkok, Thailand. Association for Computational Linguistics.

Appendix Contents

A Data Details
B Implementation Details
C Training Methodology Ablation Study
D IndicGenBench Downstream Tasks
E Evaluation Metrics: Formulation and Examples
F Embedding Drift Analysis
G LLM-as-a-Judge Details
H Cross-Vocabulary Embedding Drift
I Prediction Alignment Analysis
J Global Model Performance Comparison
K Global Prediction Alignment (Recall)
L Gemma2-9B Detailed Results
M Qwen2.5-7B Detailed Results
N Cross-Metric View of Strongest Configurations
O Non-Latin Script Examples
P Artifacts
Q Model Checkpoints
R AI Assistants

A Data Details

This appendix provides detailed specifications, source distributions, and preprocessing steps for the datasets used in this work. See Table 9 for the links to the sources and their respective license.

Sampling Protocol. For each (model, language) configuration, the training data is sampled once per training size S and reused across all $|\Delta V|$ conditions, so that vocabulary-expansion effects are not confounded with sampling variability. Samples are drawn uniformly at random without replacement from the corresponding monolingual and parallel corpora. The same sample is used in both Stage 1 and Stage 2 of training.

Sanskrit Monolingual. For the monolingual Sanskrit, we utilized the Sanskrit subset of the **CC-100** corpus (Conneau et al., 2020; Wenzek et al., 2020) from the University of Edinburgh's StatMT project. The subset contains 2,049,884 text samples (≈367 MB).

Sanskrit Parallel. For alignment, we used the **Saamayik** Sanskrit–English Parallel Corpus (Maheshwari et al., 2024). Unlike traditional corpora, Saamayik emphasizes contemporary prose. The dataset contains approximately 53,000 sentence pairs derived from: **Spoken Tutorials (Technical)** (≈23,835 pairs of instructional content); **NIOS (Educational)** (11,356 pairs of curriculum content); **Bible Translations** (7,838 pairs); **Gitasopanam** (5,885 pairs of spiritual texts).

Tibetan Monolingual. For the monolingual Tibetan, we utilized a Classical Tibetan corpus of canonical, para-canonical, and non-canonical scriptures and non-scriptures. The corpus was compiled as part of the broader research project of which this work is a part; its publicly available components include the **spsither Tibetan monolingual corpus** (Phuntsog, 2024) and **MC2 Corpus** (Zhang et al., 2024). The spsither repository is published openly on HuggingFace without an explicit license declaration; we used it solely for academic research. The full corpus construction and preprocessing details are provided in Bilitski et al. (2026).

Tibetan Parallel. For Tibetan–English alignment, we utilized the **Tibetan to English Translation Dataset** (Moore, 2024), scraped from *Lotsawa House*[4] and released under the CC BY-NC 4.0 license.

Data Release. The third-party datasets used in this work are publicly available at the URLs listed in Table 9, under the licenses indicated there, and were used in accordance with their stated terms for research purposes. The Tibetan corpus compiled within the broader research project is not redistributed as part of this work; the publicly available components are accessible at their original locations as listed in Table 9.

B Implementation Details

We utilize the HuggingFace Transformers library[5] for all experiments. The main hyperparameters used for the Continual Pre-Training (CPT) stages are detailed in Table 2; all remaining settings are available in the project's repository.

C Training Methodology Ablation Study

To compare two training configurations for integrating new tokens, we conduct an ablation study using Gemma2-2B with 100k Tibetan training samples and 128 new tokens ($|\Delta V| = 128$). In the first configuration (**new embeddings**), we freeze all pre-trained parameters and update only the newly initialized embeddings (E'_{in} and E'_{out}). In the second configuration (**Full Embedding Matrix**), we continue training from the first stage but unfreeze the entire embedding matrix, allowing all token embeddings to be updated while keeping the transformer layers frozen.

Table 3 presents the results of this comparison across multiple learning rates. We observe a consistent degradation in translation quality (measured

[4] https://www.lotsawahouse.org/

[5] https://huggingface.co/docs/transformers/en/index

Hyperparameter	Value
Optimizer	AdamW
Learning Rate	2×10^{-4}
Scheduler	Cosine Warmup
Warmup Ratio	0.03
Max Sequence Length	512
Batch Size	
Gemma2-2B / Qwen2.5-1.5B	8
Gemma2-9B / Qwen2.5-7B	32
Training Epochs	
Stage 1 (new embeddings)	2
Stage 2 (LoRA CPT)	2
LoRA Configuration	
Rank (r)	8
Alpha (α)	16
Dropout	0.05
Target Modules	All Linear Layers
Compute (GPUs)	
Gemma2-2B / Qwen2.5-1.5B	1× NVIDIA RTX 3090
Qwen2.5-7B	1× NVIDIA A100
Gemma2-9B	1× NVIDIA B200

Table 2: Hyperparameters for Stage 1 (Embedding Alignment) and Stage 2 (LoRA Adaptation) across all model scales.

by chrF++) when the full embedding matrix is unfrozen. This suggests that modifying the pre-trained embedding space disrupts the model's existing internal alignment, whereas restricting updates to the new tokens allows them to align effectively with the frozen semantic space. Based on these findings, we adopt the **new embeddings** strategy for all subsequent experiments.

Learning Rate	New Embeddings (chrF++)	Full Embedding Matrix (chrF++)
3e-05	**17.38**	15.31
5e-05	**17.11**	14.09
8e-05	**17.38**	14.40
5e-05*	**17.40**	13.39
1e-03	**17.46**	14.05

Table 3: Ablation study results (chrF++) comparing the standard approach (updating only new embeddings) versus continuing training by unfreezing the Full Embedding Matrix. Performance consistently degrades when the full matrix is unfrozen. *Cosine scheduler with no warmup, versus `constant_with_warmup` with 25 warmup steps in the second row.

D IndicGenBench Downstream Tasks

This appendix provides detailed information regarding the evaluation using IndicGenBench suite (Singh et al., 2024). All prompts were formulated after multiple experiments and adjusted to the specific requirements of each task, while maintaining a consistent pattern across all models.

Machine Translation: IndicGenBench's FLORES-IN extends FLORES-200 (NLLB Team et al., 2024) to 29 Indic languages, re-using FLORES-200 data for the 22 languages it already covers—including Tibetan and Sanskrit—and collecting human translations for 7 additional languages. We evaluate on the FLORES-IN devtest split (1,012 parallel sentences per language) and report results using `chrF++` score. Since our study targets the effect of vocabulary expansion rather than absolute translation quality, we report `chrF++` as relative change from each base model. The exact prompt provided to the pre-trained models is shown in Figure 11.

XorQA-IN: Cross-lingual question answering based on XorTYDI (Asai et al., 2021), which contains questions in target languages paired with English evidence passages. The model must provide the answer in the target language. The answer is compared to an annotated answer, translated by human translators. Results are reported using `chrF++` score. The base models, Gemma2-2B and Qwen2.5-1.5B, achieve 12.0 and 5.0 for Tibetan, and 17.9 and 7.3 for Sanskrit, respectively. For comparison, IndicGenBench reports scores of only 0.9 for Tibetan and 11.0 for Sanskrit using the larger Gemma-7B-IT model. The exact prompt provided to the models is shown in Figure 12. Evaluation was performed using *Language Model Evaluation Harness* (Gao et al., 2024).

CrossSum-IN: Cross-lingual summarization task based on CrossSum (Bhattacharjee et al., 2023), which contains English news articles. The task is to summarize the English article into the target language. The summary is compared to an annotated summary, translated by human translators. Results are reported using `chrF++` score. The base models, Gemma2-2B and Qwen2.5-1.5B, achieve 0.0 and 0.02 for Tibetan, and 8.2 and 13.9 for Sanskrit, respectively. Similarly, IndicGenBench reports scores of 2.8 for Tibetan and 13.5 for Sanskrit for Gemma-7B-IT. The exact prompt provided to the models is shown in Figure 13. Evaluation was performed using *Language Model Evaluation Harness* (Gao et al., 2024).

Model input

User prompt
Translate to {*target language*}.
Output only the translation.
{*source language*}: {*text*}
{*target language*}:

Figure 11: The prompt used for Machine Translation.

Model input

System prompt
You are a helpful assistant that answers questions based on context. Your response must contain ONLY the answer text, with no prefixes like 'The answer is' or 'Answer:'.

User prompt
Read the following context and answer the question in Tibetan.
Context: *Context*
Question: *Question*
Answer in Tibetan (Return ONLY the answer)

Figure 12: The prompt used for XorQa-IN.

Model input

System prompt
You are a helpful assistant that summarizes text.

User prompt
Read the following English article and write a brief summary in Tibetan.
Article: {*Article*}
Summary in Tibetan:

Figure 13: The prompt used for CrossSum-IN.

We find that the results for XorQA and CrossSum are consistently low across all configurations—often underperforming the base models—and lack a clear trend. We suggest that these tasks are sufficiently complex, demanding both cross-lingual alignment and instruction-following capabilities, which fall outside the scope of our current methodology. While minor patterns may be observed, we defer a deeper analysis to future work on both cross-lingual capabilities and supervised fine-tuning in LRLs.

E Evaluation Metrics: Formulation and Examples

This appendix provides the formal definitions of our evaluation metrics, illustrated on a short synthetic sentence of 8 tokens. We assume a toy expansion of $|\Delta V_{\text{toy}}| = 2$ tokens, $\Delta V_{\text{toy}} = \{t'_1, t'_2\}$, and use $t_a, t_b, t_c \in V$ for tokens from the original vocabulary. The tokenized sentence and the model's top-1 predictions are shown in Table 4. The same procedure applies to real ΔV sizes and to the FLORES-IN test set used in the main experiments.

E.1 Token Usage

Token usage measures the fraction of positions in the tokenized output whose token is drawn from the expanded vocabulary ΔV. Formally, for a tokenized sequence $[t_1, \ldots, t_n]$:

$$\text{Token Usage}(\Delta V) = \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}[t_i \in \Delta V]. \quad (2)$$

For the running example (Table 4), 3 of the 8 positions are tokens in ΔV, giving a per-sentence token usage of $3/8 = 37.5\%$. The values reported in Figure 6 are obtained by aggregating this quantity over the full FLORES-IN test set.

E.2 Embedding Drift

Embedding drift measures the cosine similarity between a new token's initialized embedding t'_{init} and its final embedding t'_{final} after training:

$$\text{CS}(t') = \frac{\langle t'_{\text{init}},\ t'_{\text{final}} \rangle}{\|t'_{\text{init}}\| \cdot \|t'_{\text{final}}\|}, \tag{3}$$

where t'_{init} is obtained via mean initialization (Eq. 1) from the embeddings of t''s subword components $t_a, t_b, t_c \in V$. Figure 14 illustrates this geometrically in a 2D space for didactic purposes, with an example of a low cosine similarity (panel a, CS = 0.42) and a high cosine similarity (panel b, CS = 0.98).

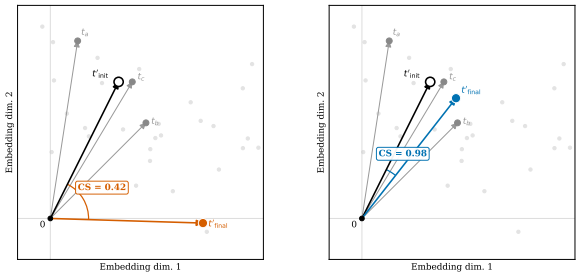


Figure 14: Illustration of embedding drift for a new token $t' \in \Delta V$. Left (panel a): low cosine similarity (CS = 0.42). Right (panel b): high cosine similarity (CS = 0.98). The token is mean-initialized as the centroid of its subword components $t_a, t_b, t_c \in V$. Coordinates are illustrative.

E.3 Prediction Alignment: Recall and Specificity

The prediction alignment metrics are computed over the positions of the tokenized reference sentence. For a sentence x tokenized as $[t_1, \ldots, t_n]$, we compute at each position $i \geq 2$ the model probability $p_\theta(\cdot \mid t_1 \ldots t_{i-1})$ over the full vocabulary $V' = V \cup \Delta V$, and take the top-1 greedy prediction:

$$\hat{t}_i = \arg\max_{v \in V'} p_\theta(v \mid t_1 \ldots t_{i-1}). \tag{4}$$

Position $i = 1$ has no prefix and is therefore used as context only. Each remaining position contributes to exactly one of the two metrics, depending on whether the gold token t_i belongs to V or ΔV. For a tokenized sentence $[t_1, \ldots, t_n]$ and its greedy predictions $[\hat{t}_2, \ldots, \hat{t}_n]$:

$$\text{Recall}(\Delta V) = \frac{\sum_{i=2}^{n} \mathbf{1}[t_i \in \Delta V \wedge \hat{t}_i = t_i]}{\sum_{i=2}^{n} \mathbf{1}[t_i \in \Delta V]}, \tag{5}$$

$$\text{Specificity}(V) = \frac{\sum_{i=2}^{n} \mathbf{1}[t_i \in V \wedge \hat{t}_i = t_i]}{\sum_{i=2}^{n} \mathbf{1}[t_i \in V]}. \tag{6}$$

We refer to this quantity as *specificity* over V by analogy with the sensitivity–specificity pairing common in classification metrics; equivalently, it is recall on the class of original-vocabulary tokens. The two denominators are disjoint and sum to $n-1$. Table 4 works through the computation on the 8-token example. Aggregating over the 7 evaluated positions:

$$\text{Recall}(\Delta V) = \tfrac{1}{3} \approx 33.3\%,$$
$$\text{Specificity}(V) = \tfrac{1}{4} = 25.0\%.$$

The values reported in Section 4 are aggregated over thousands of positions across the full FLORES-IN test set.

Pos.	Gold t_i	Set	Pred. $\hat{t}_i$	Match?
1	t_a	V	-	-
2	t'_1	ΔV	t_a	×
3	t_b	V	t_c	×
4	t'_1	ΔV	t'_1	✓
5	t_c	V	t_c	✓
6	t'_2	ΔV	t_a	×
7	t_a	V	t_b	×
8	t_b	V	t_a	×
ΔV: 1/3 correct ⇒ Recall = 33.3%				
V: 1/4 correct ⇒ Specificity = 25.0%				

Table 4: Per-position prediction outcomes on the running synthetic example. Position 1 is used as context only and excluded from both metrics.

E.4 LLM-as-a-Judge

For open-ended generation, the LLM judge receives a Tibetan sentence prefix and a model-generated continuation, and returns a score on a five-point scale (Very bad=1 to Very good=5) along with a brief reasoning. Figure 15 shows an example input and the expected output format. Full protocol details, including the system prompt, scoring rubric, and human validation, are provided in Appendix G.

Input prefix:

དེ་ནས་ཕྱི་རོལ་ན་གནས་པའི་བྱམ་པའི་ཚུས་བདག་ཉིད་ལ་དབང་བསྒྱུར

Generated continuation:

། དེ་ནས་བདག་ཉིད་ཀྱི་ཡུལ་ལ་དབང་བསྒྱུར་བས། ཐུགས་ཀྱི་དབང་བསྒྱུར་རོ། །ཚོར་སྐད་ཀ་ཐ་ཀ་ཚ་ཏི་ཏིཏྠཱ། ཞེས་བརྗོད་པས་བྱམ་པ་ལས་བྱང་བའི་བྱམ་པའི་ཚུ་མངོན་པར་འགྱུར་པར་བྱ

Expected judge output (JSON):

```
{
"Reasoning": "<brief explanation>",
 "Score": "<one of: Very bad, Bad,
      Neutral, Good, Very good>"
}
```

Figure 15: Example input to the LLM-as-a-Judge: a Tibetan prefix, the model's continuation, and the expected JSON output format. Translation and transliteration are provided in Appendix O.

F Embedding Drift Analysis

This appendix provides the detailed analyses referenced in the Embedding Drift paragraph of Section 4.3.1. We examine the behavior of new embeddings in Gemma2-2B and Qwen2.5-1.5B from three complementary perspectives: (i) cosine similarity to initialization, separated by whether the new token appears in the tokenized training set, (ii) embedding norms before and after training, and (iii) pairwise similarities among the new embeddings. All analyses are performed at $|\Delta V| = 8{,}192$ with 100k training samples. We report medians throughout, consistent with the embedding drift metric used in the main results and to avoid distortion from tail values.

Seen and Unseen Tokens. We define a new token as *seen* if it appears at least once in the tokenized training set, and as *unseen* otherwise. Seen tokens act as cross-entropy targets at least once during training and have the opportunity to be learned from those steps. Unseen tokens, by contrast, never act as targets; the only updates their embeddings receive come from being non-targets at other steps. Because the model's output is a probability distribution over the full vocabulary, increasing the probability of the actual target at a given step requires reducing the probability of every other token, including unseen new tokens. This indirect effect is small at any single step, but accumulates across training and shapes the embeddings of rarely or never-targeted tokens, a regime studied in prior work on under-trained representations (Biś et al., 2021; Yu et al., 2022; Land and Bartolo, 2024). We use this distinction as a diagnostic. If embeddings drift because tokens are being trained as targets, seen tokens should drift more than unseen ones; if seen and unseen tokens drift similarly, the drift must come from another source.

Cosine Similarity by Exposure. In Gemma2-2B, drift tracks exposure: the median CS of unseen tokens is 0.97 (Tibetan) and 0.98 (Sanskrit), substantially higher than the seen-token CS of 0.78 (Tibetan) and 0.84 (Sanskrit). Seen tokens move substantially from their initialization, while unseen tokens stay close to it.

In Qwen2.5-1.5B, the seen–unseen difference observed for Gemma2-2B is absent: unseen tokens reach median CS of 0.16 (Tibetan) and 0.11 (Sanskrit), compared to seen-token CS of 0.13 (Tibetan) and 0.14 (Sanskrit). Unseen tokens drift to nearly the same extent as seen ones, meaning that CS to initialization in Qwen2.5-1.5B does not separate trained from untrained new tokens.

Table 5(a) summarizes these results across the four model-language configurations.

Embedding Norms. As a complementary measurement, we report the L2 norms of new embeddings before and after training (Table 5(b)). In both models, seen tokens grow substantially in norm (ratios 1.16–1.25), while unseen tokens stay close to their initialization (ratios 0.90–0.97). This is consistent with prior work using embedding norm as an indicator of under-trained tokens, whose embeddings remain near their initialization or decay toward zero (Land and Bartolo, 2024), and gives a separate test of whether target-side updates are reaching the new tokens. The pattern holds across both languages and both architectures, including Qwen2.5-1.5B, where the seen–unseen distinction was absent in CS.

Pairwise Similarity Among New Embeddings. The CS and norm analyses describe how each added embedding moves *relative to its own initialization*. A complementary question is *where* the embeddings end up relative to one another: do they spread across the embedding space, or do they collapse toward a shared region? To answer this, we compute the pairwise cosine similarity between distinct pairs of new embeddings after training. If the new embeddings spread evenly, their pairwise CS should be close to the values observed for random pairs of base-vocabulary embeddings of the same model. If, instead, the new embeddings cluster together, their pairwise CS will be substantially

			(a) Cosine similarity to initialization			(b) L2 norms before and after training		
Model	**Lang**	**Status**	n	%	**CS (med)**	**init**	**final**	**ratio**
Gemma2-2B	Tibetan	unseen	772	9.4%	0.97	0.83	0.80	0.96
		seen	7,420	90.6%	0.78	0.84	1.05	1.25
Gemma2-2B	Sanskrit	unseen	3,982	48.6%	0.98	1.06	1.03	0.97
		seen	4,210	51.4%	0.84	1.10	1.28	1.16
Qwen2.5-1.5B	Tibetan	unseen	1,840	22.5%	0.16	0.55	0.52	0.95
		seen	6,352	77.5%	0.13	0.57	0.68	1.19
Qwen2.5-1.5B	Sanskrit	unseen	607	7.4%	0.11	0.59	0.53	0.90
		seen	7,585	92.6%	0.14	0.63	0.75	1.19

Table 5: Geometry of new embeddings at $|\Delta V| = 8{,}192$ after training on 100k samples, classified as *seen* if the token appears in the tokenized training set and *unseen* otherwise. **(a)** Median cosine similarity between initial and final embeddings; n = count; % = share of the 8,192 new tokens. **(b)** Median L2 norms at initialization and after training; *ratio* = final / init.

higher than this reference.

To estimate pairwise CS, we draw 500 random pairs of new embeddings and compare them to 500 random pairs from the base vocabulary of the same model, used as a reference for "typical" pairwise CS in that embedding space. Table 6 reports the median pairwise CS in each setting and the ratio between the added and base values.

Model	**Lang**	**new**	**base**	**ratio**
Gemma2-2B	Tibetan	0.41	0.16	2.5×
Gemma2-2B	Sanskrit	0.17	0.15	1.1×
Qwen2.5-1.5B	Tibetan	0.55	0.10	5.5×
Qwen2.5-1.5B	Sanskrit	0.47	0.10	4.6×

Table 6: Median pairwise cosine similarity between 500 random pairs of new embeddings (*new*) and 500 random pairs of base-vocabulary embeddings of the same model (*base*). The *ratio* column is (*new*) / (*base*).

In Gemma2-2B, new embeddings show pairwise CS comparable to the base reference for Sanskrit ($1.1\times$) and moderately higher for Tibetan ($2.5\times$). In Qwen2.5-1.5B, the ratio is substantially larger: $5.5\times$ for Tibetan and $4.6\times$ for Sanskrit, indicating that the new embeddings collapse toward a narrow region of the embedding space. This pattern matches the representation degeneration problem (Gao et al., 2019), in which embeddings cluster together rather than spreading across the embedding space, and which Biś et al. (2021) and Yu et al. (2022) link to rare and under-trained tokens.

Summary. The three perspectives describe a consistent picture for Qwen2.5-1.5B. CS to initialization in this model does not separate seen from unseen new tokens, although embedding norms do, indicating that target-driven updates are still active. The pairwise-similarity analysis locates the destination of the drift in Qwen2.5-1.5B: the new embeddings collapse toward a narrow region of the embedding space, consistent with representation degeneration. In Gemma2-2B, by contrast, CS, norms, and pairwise similarity all separate seen from unseen tokens, and new embeddings remain comparatively spread out. Taken together, these results indicate that embedding drift measured by CS is a useful indicator of adaptation for Gemma2-2B but not for Qwen2.5-1.5B. A more complete account of the architectural and training factors that produce the Qwen2.5-1.5B behavior—including output embedding geometry, weight tying, and base-vocabulary statistics—requires controlled experiments and falls outside the scope of this work.

G LLM-as-a-Judge Details

We utilize Claude-4.5-Sonnet as an automated judge to evaluate open-ended generation quality. To ensure statistical stability, we perform 5 iterations per sample at a temperature of 0.8. The final score for each model is the average across all samples and iterations.

The judge evaluates the quality of Tibetan text continuations using a five-point scale (Very bad, Bad, Neutral, Good, Very good) based on three primary criteria:

1. **Coherence**: Logical flow from the input text.
2. **Grammar & Syntax**: Correct and natural language usage.
3. **Relevance**: Adherence to the topic of the prefix.

The five-point scale was mapped to numerical values (1–5) to calculate final performance met-

rics. The exact system prompt provided to the LLM judge is shown in Figure 17. To validate the LLM judge scores, we compare them to ratings from six Tibetan language experts, scholars who research Tibetan texts and were collaborators on the broader research project of which this work is a part. Each expert independently rated the same 50 model outputs generated by a single model configuration (Gemma2-2B with 128 new tokens, trained on 100k samples). The experts followed the same evaluation criteria and five-point scale used by the LLM judge (verbal labels mapped to numerical values 1–5), and were not informed of the LLM judge scores. The exact instructions provided to the experts are shown in Figure 16. The correlations between the averaged human ratings and the LLM judge scores are Pearson 0.63 ($p < 10^{-6}$) and Spearman 0.54 ($p < 10^{-4}$), indicating moderate alignment with human ratings.

H Cross-Vocabulary Embedding Drift

Figure 18 tracks cosine similarity for the top-512 new tokens across different expansion sizes. The y-axis lists these tokens; greyed-out regions correspond to the $|\Delta V|$ values at which a given token has not yet been added to the expansion set. The x-axis varies the total vocabulary size $|\Delta V|$, so each row shows how a fixed token drifts depending on how many other tokens are added alongside it. For readability, only a sample of tokens is labelled on the y-axis; these are illustrative reference points and not the full list.

For Tibetan (left), CS values increase from left to right: the same tokens drift less as $|\Delta V|$ grows, consistent with each new token appearing in training less often as $|\Delta V|$ grows under a fixed amount of data. For Sanskrit (right), the pattern is less regular. Large vertical bands of high CS (near 1.0) persist across all expansion sizes, indicating that many tokens remain near their initialization regardless of $|\Delta V|$. There is no clear monotonic gradient across the x-axis; some tokens show lower CS in larger vocabularies than in smaller ones, reinforcing the limitation of frequency-based vocabulary construction for Sanskrit noted in §4.2.

I Prediction Alignment Analysis

This appendix provides the per-configuration prediction alignment figures for Gemma2-2B referenced in §4.2. Figure 19 shows recall over ΔV as a function of expansion size and training data, separately for Tibetan and Sanskrit. Figure 20 shows the corresponding specificity values for original-vocabulary tokens.

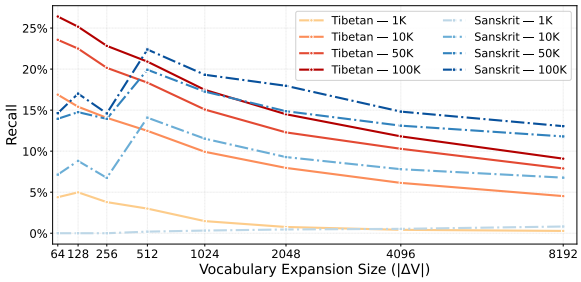


Figure 19: Gemma2-2B: Recall over ΔV across different configurations.

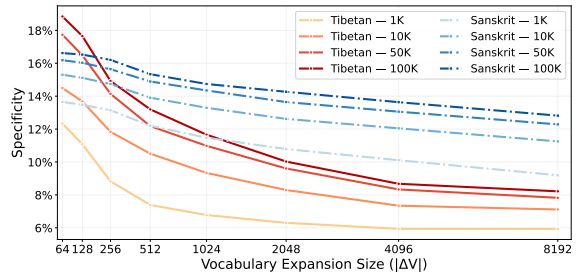


Figure 20: Gemma2-2B: Specificity over V across different configurations.

The recall curves confirm the asymmetry reported in §4.2: Tibetan recall declines monotonically with expansion size in higher data regimes, whereas Sanskrit recall shows no consistent trend, with comparable values at $|\Delta V| = 64$ and $|\Delta V| = 8{,}192$. The specificity curves show a clear contrast: in Tibetan, specificity drops as the vocabulary expands (from 18.8% to 8.2% in the 100k regime), indicating that predictions on original-vocabulary tokens degrade as more tokens are added. In Sanskrit, specificity remains stable across expansion sizes (12.8%–16.6%), consistent with the limited integration of new Sanskrit tokens reported in the body.

J Global Model Performance Comparison

Figure 22 compares Machine Translation performance (`chrF++` improvement) across the small Gemma2-2B and Qwen2.5-1.5B models, both languages, and all vocabulary expansion configurations.

First, Qwen2.5-1.5B exhibits balanced base capabilities across Sanskrit (11.48) and Tibetan (10.60), whereas Gemma2-2B shows a marked disparity between the two languages (15.61 vs. 5.17, respectively). This points to a more balanced cov-

Human Annotator Instructions

Your task is to evaluate the quality of a Unicode text continuation based on three criteria:

1. **Coherence**: Does the continuation follow logically from the input text?
2. **Grammar & Syntax**: Is the language correct, natural, and error-free?
3. **Relevance**: Does the content stay on topic?

IMPORTANT: Some texts may be cut off mid-sentence or contain repetition due to generation limits. Do NOT deduct points for cut-off sentences or repetitive patterns—focus only on the three criteria above.

Scoring Criteria (be strict and critical):

- **Very bad**: Bad coherence (not following the start of the sentence), bad grammar & syntax (not in the same language, misspelled words, etc.), not relevant.
- **Bad**: Same language but fails on TWO of three criteria (coherence, grammar/syntax, or relevance).
- **Neutral**: Same language with acceptable grammar, but lacks either strong coherence OR clear relevance.
- **Good**: Same language, good grammar and syntax, stays on topic AND mostly follows logical flow, but may have minor imperfections.
- **Very Good**: EXCEPTIONAL quality—perfect grammar, flawless logical continuation, highly relevant, and reads completely naturally.

Figure 16: Instructions provided to the Tibetan language experts for validating the LLM-as-a-Judge scores.

erage of the two scripts in the Qwen2.5-1.5B base vocabulary.

Second, the architectures respond differently to vocabulary expansion. For Tibetan, Qwen2.5-1.5B displays a moderate performance peak at intermediate expansion sizes (around 512 tokens in higher data regimes), followed by relatively stable gains even at larger volumes. In contrast, Gemma2-2B shows sharper degradation as expansion grows. For Sanskrit, the architectural difference becomes more pronounced. Qwen2.5-1.5B reaches its peak at small expansion sizes ($|\Delta V| \leq 256$) in the 50k–100k regimes, after which performance drops sharply and turns negative at larger expansions (down to -28.7% at $|\Delta V| = 4{,}096$, 50k samples; -47.8% at $|\Delta V| = 2{,}048$, 10k samples). Gemma2-2B, in contrast, shows scattered improvements (+10% to +22%) across most expansion sizes with no clear peak or degradation pattern, and remains positive throughout. Higher data regimes (50k–100k) improve Sanskrit performance in both models, but do not prevent the degradation observed in Qwen2.5-1.5B at large expansions. This contrast is consistent with the difference in Sanskrit baseline scores (Gemma2-2B: 15.61; Qwen2.5-1.5B: 11.48), suggesting that the architectures start from different levels of base coverage for Sanskrit and respond to expansion accordingly.

Overall, the global heatmap shows that the trade-offs between vocabulary expansion and downstream translation performance differ between the two model families, consistent with the multiple factors distinguishing them discussed in §4.3.1 and the Limitations.

K Global Prediction Alignment (Recall)

Figure 24 compares recall over ΔV across the small Gemma2-2B and Qwen2.5-1.5B models, both languages, and all vocabulary expansion configurations. In Tibetan, Qwen2.5-1.5B consistently achieves higher recall scores ($> 30\%$) than Gemma2-2B, indicating stronger adoption of the new vocabulary. In Sanskrit, Qwen2.5-1.5B shows higher recall at smaller expansion sizes and larger

Model input

System prompt

You are a professional linguistic evaluator specializing in Tibetan text analysis.

TASK: Evaluate the quality of a text continuation based on three criteria: Coherence, Grammar & Syntax, and Relevance.

IMPORTANT: Some texts may be cut off mid-sentence or contain repetition due to generation limits. Do NOT deduct points for cut-off sentences or repetitive patterns - focus only on the three criteria above.

SCORING CRITERIA:

- **Very bad**: NOT in Tibetan OR fails on all three criteria.
- **Bad**: In Tibetan but fails on TWO of three criteria.
- **Neutral**: Acceptable grammar, but lacks strong coherence or relevance.
- **Good**: Good grammar, stays on topic, and mostly follows logical flow.
- **Very Good**: Perfect grammar, flawless logical continuation, and highly relevant.

Respond ONLY with valid JSON in this exact format:

```
{
   "Reasoning": "<brief explanation in English>",
   "Score": "<one of the 5 options>"

}
```

User prompt

Sentence: *Sentence*

Figure 17: The prompt used for the LLM-as-a-judge evaluation part.

training volumes. Conversely, Gemma2-2B shows lower recall at smaller expansion sizes, reaching its peak only at 100k samples and 512 new tokens; at this configuration, MT performance reaches only 10.1%, well below the 21.7% peak (50k samples, 64 tokens)—reinforcing that recall and MT do not necessarily track each other.

L Gemma2-9B Detailed Results

Machine Translation Performance. Figure 26 shows MT results that largely reinforce the trends observed in the 2B version. While the Tibetan base model performs slightly better than its smaller counterpart, the Sanskrit base model starts with a substantially lower score (7.56 vs. 15.61), trailing Gemma2-2B by over 50%.

In Tibetan, a clear performance drop is observed for $|\Delta V| \geq 1{,}024$ across all sample sizes. Once the training volume reaches 50k samples or more, the best results are consistently achieved at $|\Delta V| = 64$ tokens. At 10k samples, $|\Delta V| = 0$ remains competitive with the expansion sets, while at 1k samples the picture is less consistent across expansion sizes. For Sanskrit, the low baseline allows for improvements of nearly 70%, with trends favoring 64/128 tokens at 10k and 100k samples. However, due to scattered results at 300k and the risk of degradation with larger vocabularies, 0 new tokens remains a stable, low-risk choice. The 1k training regime is an exception: for Sanskrit, it yielded performance below the base model in most expansion sizes.

Embedding drift. When comparing the training embedding drift for the 100k models in Fig-

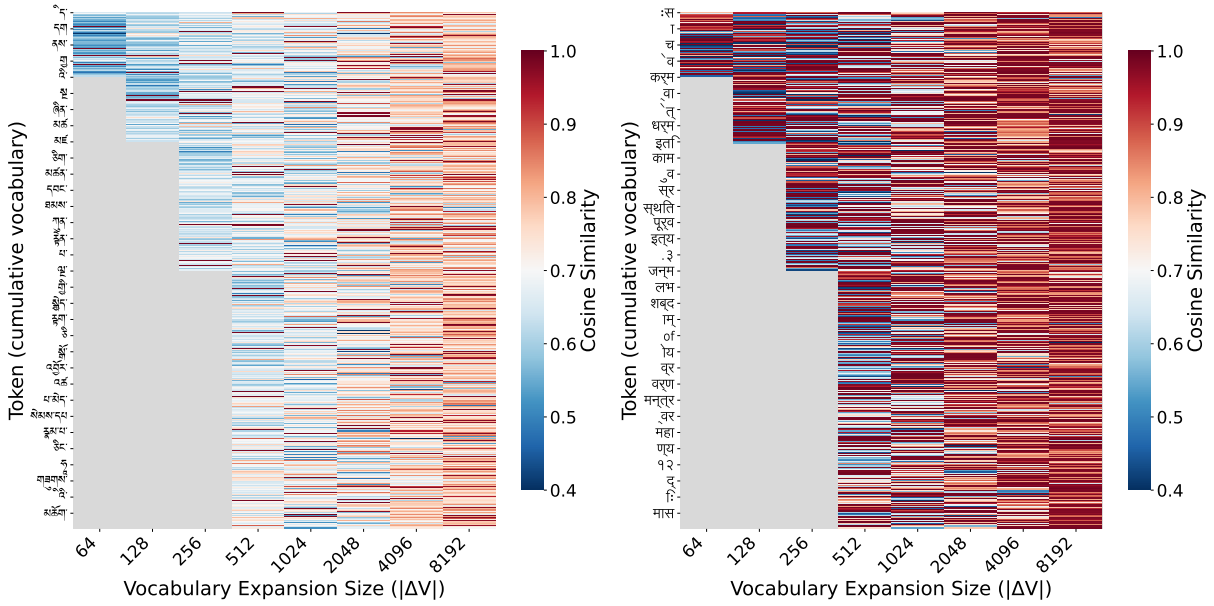


Figure 18: Heatmap of Cosine Similarity for the first 512 tokens across different expansion sizes in Gemma2-2B Tibetan (left) and Sanskrit (right).

ure 25, we observe similar trends, though the cosine similarity values for Gemma2-9B are consistently higher than those for Gemma2-2B. Since Stage 1 training was conducted independently of internal parameters, the model size appears to influence the adaptation of new embeddings. Specifically, larger models may require more training steps for effective adaptation. This is supported by the 300k pre-trained Gemma2-9B results, where the median values reach lower scores than 100k results and are slightly lower than the 100k pre-trained Gemma2-2B scores.

M Qwen2.5-7B Detailed Results

Figure 21 reports MT performance for Qwen2.5-7B on Tibetan. Three observations distinguish this scale from the smaller models. First, with a strong baseline (18.08 chrF++), improvements over the base model appear only at $S \geq 50$k samples; smaller training volumes fall below baseline across all expansion sizes. Second, the only consistent trend across $|\Delta V|$ is that the smallest expansions ($|\Delta V| \in \{0, 64\}$) yield the lowest scores at 50k and 100k samples—the opposite of the pattern observed in the other models, where small expansions are typically the safest choice. Third, the gains from any non-trivial expansion size are modest and largely uniform; given that efficiency improves with $|\Delta V|$ (§ 4.1), larger expansions may be a reasonable choice for this model when the efficiency-performance balance is considered.

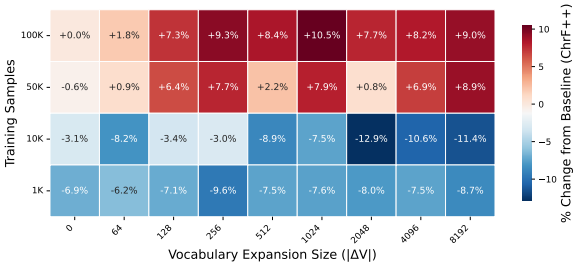


Figure 21: Qwen2.5-7B: MT Performance, English → Tibetan, % improvement over base (chrF++ 18.08).

N Cross-Metric View of Strongest Configurations

Figure 23 summarizes, for each combination of metric, language, model family, and model size, the configuration $(|\Delta V|, S)$ that yields the strongest result. Marker color denotes the language (Tibetan / Sanskrit), shape denotes the metric, fill style denotes the model family (filled for Gemma, open for Qwen), and marker size denotes the model scale (small for 2B/1.5B, large for 9B/7B).

Scope and disclaimers. Several constraints shape the contents of this figure and should be considered when interpreting it. **(i)** Not all models are evaluated on all metrics and languages. In particular, Qwen2.5-7B is trained and evaluated only on Tibetan, as noted in §4.3. **(ii)** Prediction alignment (recall and specificity) and embedding drift are not defined at $|\Delta V| = 0$, since these metrics require the existence of new tokens; configurations with no expansion are therefore excluded from these

metrics by construction. **(iii)** Token usage is omitted, as it is monotonic in $|\Delta V|$ by definition and therefore uninformative under a "strongest configuration" view. **(iv)** The LLM-as-a-Judge metric is reported only for Gemma2-2B on Tibetan with $S = 100$k, following the protocol described in §3.2. **(v)** Gemma2-9B is the only model trained with $S = 300$k; the absence of other models from this row reflects experimental scope rather than performance. **(vi)** For embedding drift, the strongest configuration corresponds to the lowest median cosine similarity, reflecting the strongest adaptation of new embeddings. For all other metrics, the strongest configuration corresponds to the highest value.

Observations. The majority of points concentrate around $|\Delta V| = 64$, consistent with the moderate-expansion regime identified in §4.1. A smaller group lies near $|\Delta V| \approx 512$ and corresponds primarily to Qwen2.5-1.5B prediction alignment configurations, in line with the observation in §4.3 that Qwen integrates new tokens more uniformly across expansion sizes. Across languages, Tibetan and Sanskrit cluster in the same $|\Delta V|$ range, indicating that the location of the strongest configuration is largely script-independent, even though the magnitude of the gains differs substantially between the two languages (§4.2). With respect to training volume, the strongest configurations are concentrated at $S = 100$k and $S = 300$k, with no points in the 1k or 10k rows, consistent with the finding that vocabulary expansion is difficult to train effectively under limited CPT data. Across metrics, machine translation, prediction alignment, and embedding drift converge on similar regions of the $(|\Delta V|, S)$ space, whereas LLM-as-a-Judge is the only metric whose strongest configuration lies at $|\Delta V| = 0$, matching the decline reported in Table 1. `chrF++` rewards character n-gram overlap with reference translations, and recall and specificity measure next-token prediction accuracy on gold sequences, whereas LLM-as-a-Judge scores open-ended continuations on coherence, grammaticality, and relevance (Appendix G); the difference between LLM-as-a-Judge and the other metrics may reflect that these are different aspects of generation quality. LLM-as-a-Judge is also evaluated on a narrower subset than the other metrics (Gemma2-2B on Tibetan with $S = 100$k only, as noted in disclaimer (iv) above), so this observation should be interpreted accordingly. A clear exception is Qwen2.5-1.5B embedding drift, whose strongest configurations are at $|\Delta V| = 8{,}192$ in both languages; this is discussed further in Appendix F.

Training data per new token. For each configuration, the ratio $S/|\Delta V|$ gives the number of CPT samples per new token. Across the configurations of Figure 22, this ratio and `chrF++` are positively rank-correlated in all four model–language pairs (Table 7).

Model	Lang	Spearman	p
Gemma2-2B	Tibetan	$+0.93$	$< 10^{-14}$
Gemma2-2B	Sanskrit	$+0.67$	$< 10^{-4}$
Qwen2.5-1.5B	Tibetan	$+0.51$	$< 10^{-2}$
Qwen2.5-1.5B	Sanskrit	$+0.55$	$< 10^{-2}$

Table 7: Spearman rank correlation between `chrF++` and $S/|\Delta V|$ for each model–language pair, over the $n = 32$ configurations with $|\Delta V| > 0$ that pool the four CPT data volumes and the eight expansion sizes.

O Non-Latin Script Examples

This appendix provides translations and transliterations of Tibetan and Sanskrit examples used in Figure 1 (cross-lingual tokenization) and Figure 15 (LLM-as-a-Judge input). Sanskrit is transliterated using the International Alphabet of Sanskrit Transliteration (IAST); Tibetan is transliterated using the Extended Wylie Transliteration Scheme (EWTS).

Figure 1 — FLORES-IN aligned sentence. English (translation): *"Some festivals have special camping areas for families with young children."*
Sanskrit (Devanagari):
कतिपय उत्सवे क्षुद्र बालकस्य परिवारं कृते विशेष शिविर क्षत्रं इति भवति ।
Sanskrit (IAST): *katipaya utsave kṣudra bālakasya parivāraṃ kṛte viśeṣa śivira kṣatraṃ iti bhavati.*
Tibetan (script):
དུས་སྟོན་འགའ་ཞིག་ལ་བུ་ཕྲུག་ཡོད་མཁན་གྱི་ཁྱིམ་ཚང་ཆེད་སྒར་བཙུག་ས་ཞི་ས་གནས་དམིགས་གསལ་བ་ཡོད།
Tibetan (EWTS): *dus ston 'ga' zhig la bu phrug yod mkhan gyi khyim tshang ched sgar btsug sa'i sa gnas dmigs gsal ba yod/*

Figure 15 — LLM-as-a-Judge input example.
Tibetan prefix (script):
དེ་ནས་ཕྱི་རོལ་ན་གནས་པའི་བུམ་པའི་ཆུས་བདག་ཉིད་ལ་དབང་བསྐུར
Tibetan prefix (EWTS): *de nas phyi rol na gnas pa'i bum pa'i chus bdag nyid la dbang bskur*
English (translation): *Then, by the water from the vase that is outside, empowerment is bestowed upon oneself.*

Tibetan continuation (script):
། དེ་ནས་བདག་ཉིད་ཀྱི་ཕུས་ལ་དབང་བསྐུར་བས། ཐུགས་ཀྱི་དབང་བསྐུར་རོ། །ཨོཾ་སརྦ་ཏ་ཐཱ
་ག་ཏ་ཨ་བྷི་ཥིཉྩ་ཧཱུྃ ཞེས་བརྗོད་པས་བུམ་པ་ལས་བྱུང་བའི་བུམ་པའི་ཆུ་མངོན་པར་འགྲུབ་པར་བྱ

Tibetan continuation (EWTS): */ de nas bdag nyid kyi lus la dbang bskur bas / thugs kyi dbang bskur ro / / oM sarba ta thA ga ta a b+hi Shiny+tsa hUM / zhes brjod pas bum pa las byung ba'i bum pa'i chu mngon par 'grub par bya*

English (translation): *Then, by bestowing empowerment upon one's body, the empowerment of mind is bestowed. By reciting "oṃ sarva tathāgata abhiṣiñca hūṃ," the vase water emerging from the vase is to be made manifest.*

P Artifacts

We detail artifacts we use and their respective usage and licenses in Table 9. For the exact versions and more details, see the *requirements.txt* file in the project's repository.

Q Model Checkpoints

In Table 8, we present the checkpoints (or snapshots) used in this work and the models' sizes and licenses.

R AI Assistants

We used AI assistants (Claude, Gemini, and ChatGPT) to support code development, phrasing suggestions, and LaTeX styling during writing. All outputs were reviewed and edited by the authors. No content was directly generated or used without human verification.

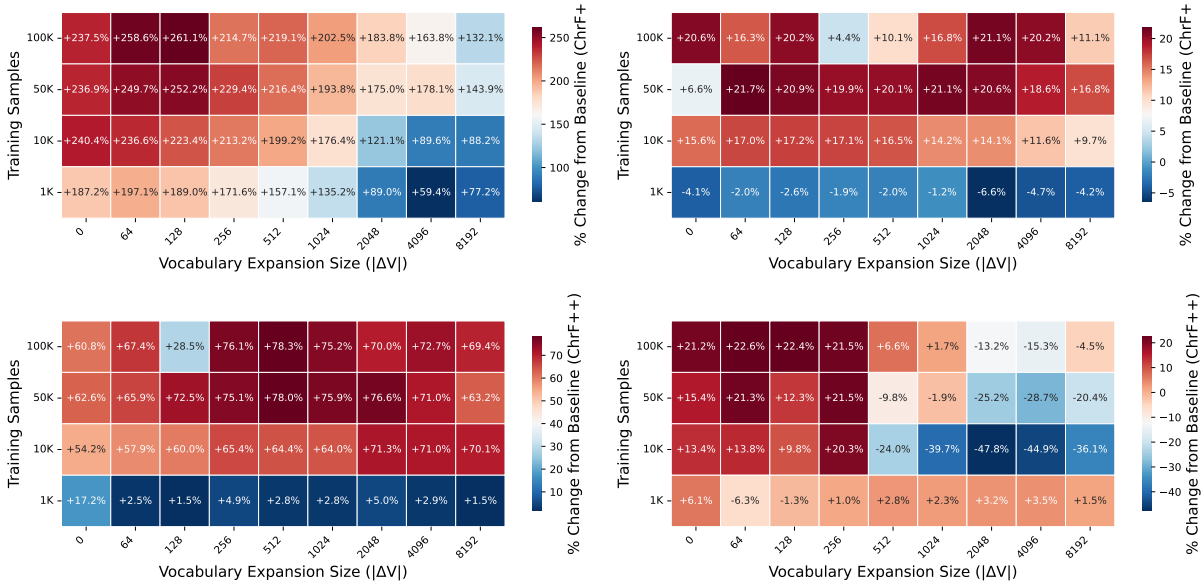


Figure 22: The grid displays results for MT Performance in chrF++ (English → Tibetan/Sanskrit), with Tibetan (left) and Sanskrit (right). The first row shows Gemma2-2B and the second row Qwen2.5-1.5B. Percentages indicate improvement relative to the base models (Gemma2-2B: Tibetan 5.17, Sanskrit 15.61; Qwen2.5-1.5B: Tibetan 10.60, Sanskrit 11.48).

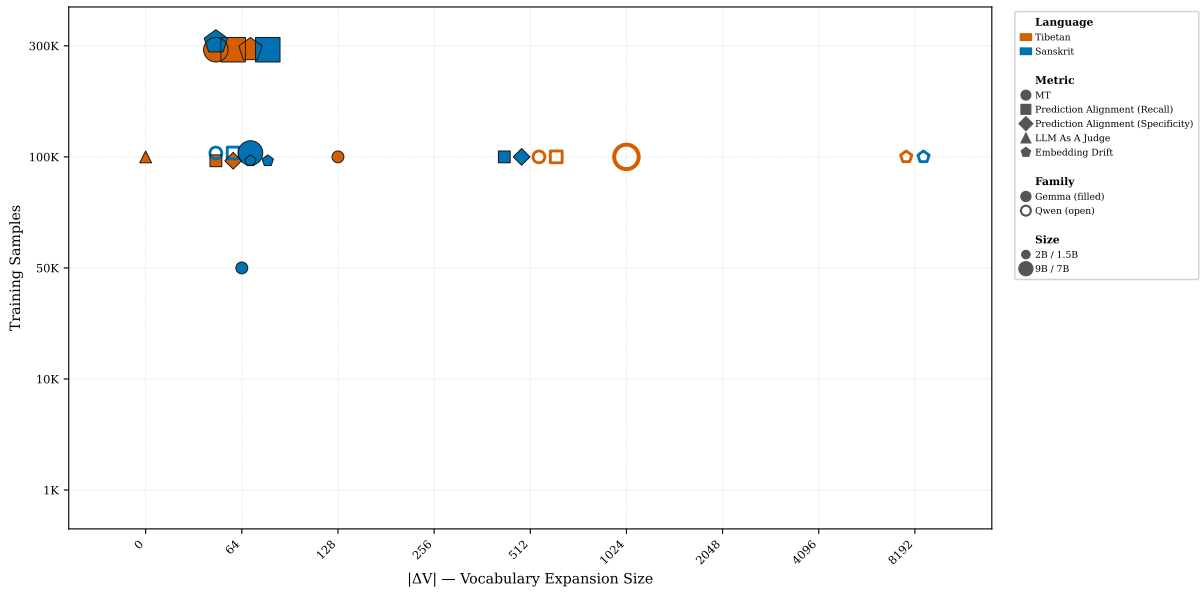


Figure 23: Configuration $(|\Delta V|, S)$ that yields the strongest result for each combination of metric, language, model family, and model size. Color denotes language, shape denotes metric, fill style denotes model family, and marker size denotes model scale.

Model	Checkpoint	License	# Param
Gemma2-2B[a]	`google/gemma-2-2b-it`	Gemma 2	2.6B
Gemma2-9B[b]	`google/gemma-2-9b-it`	Gemma 2	9B
Qwen2.5-1.5B[c]	`Qwen/Qwen2.5-1.5B-Instruct`	Apache-2.0	1.5B
Qwen2.5-7B[d]	`Qwen/Qwen2.5-7B-Instruct`	Apache-2.0	7.6B

[a] https://huggingface.co/google/gemma-2-2b-it
[b] https://huggingface.co/google/gemma-2-9b-it
[c] https://huggingface.co/Qwen/Qwen2.5-1.5B-Instruct
[d] https://huggingface.co/Qwen/Qwen2.5-7B-Instruct

Table 8: Checkpoints used during experiments, their license, and their number of parameters.

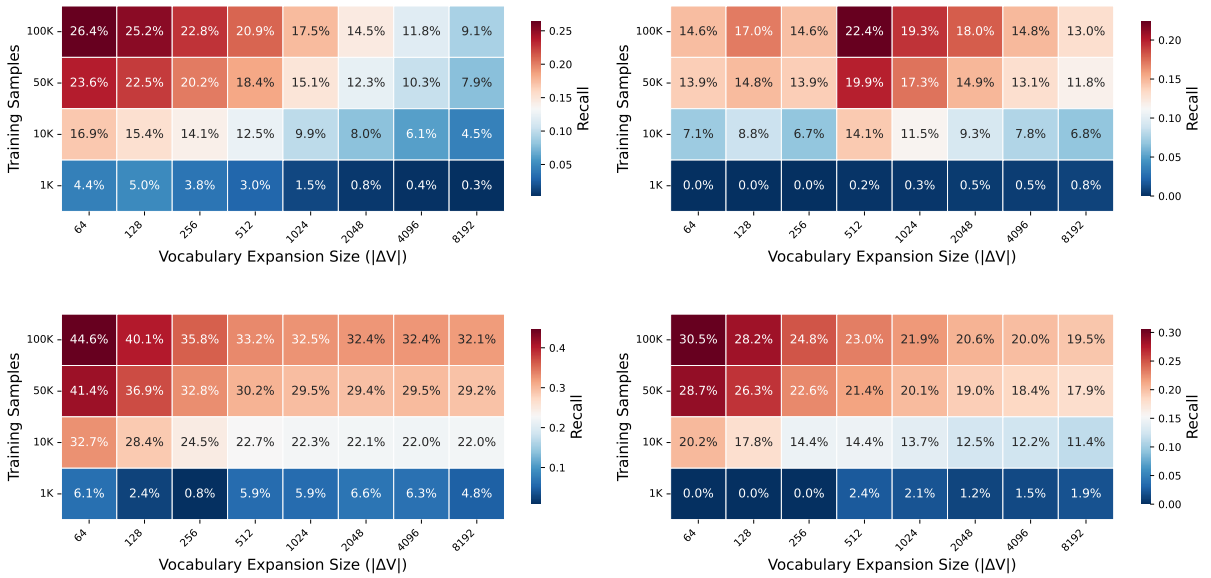


Figure 24: Prediction Alignment (Recall). Heatmap showing the recall of new tokens. The grid displays results for Tibetan (left) and Sanskrit (right), with the first row showing Gemma2-2B and the second row Qwen2.5-1.5B.

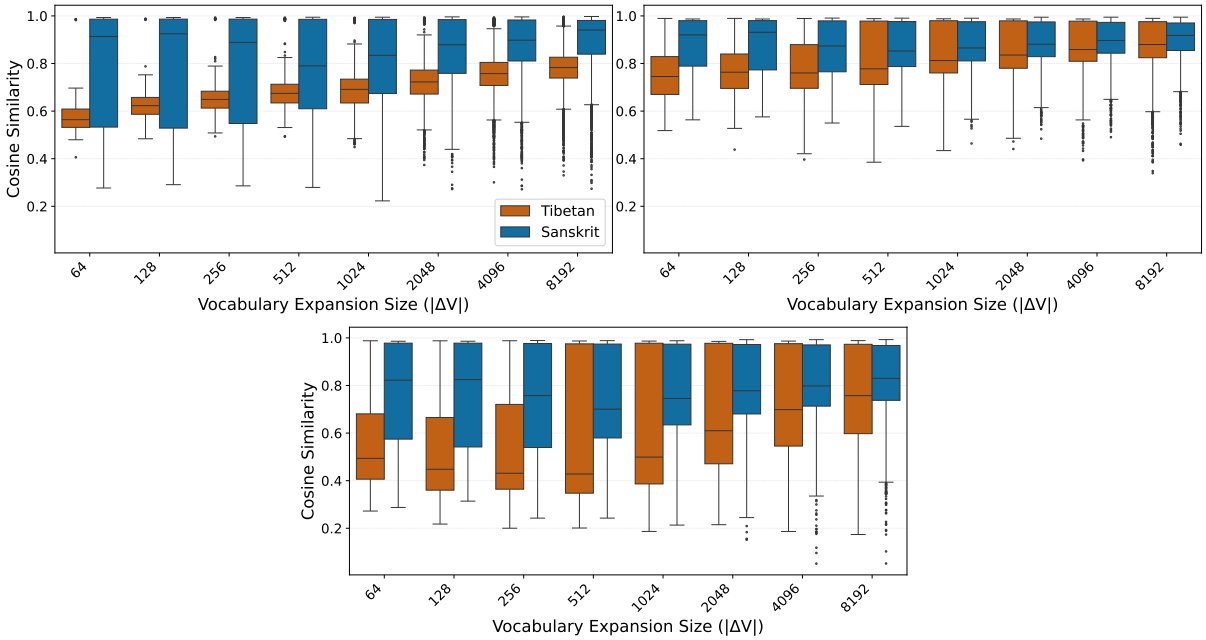


Figure 25: Embedding drift comparison between Gemma2-2B (top-left) and Gemma2-9B (top-right), trained on 100k samples; bottom: Gemma2-9B trained on 300k samples.

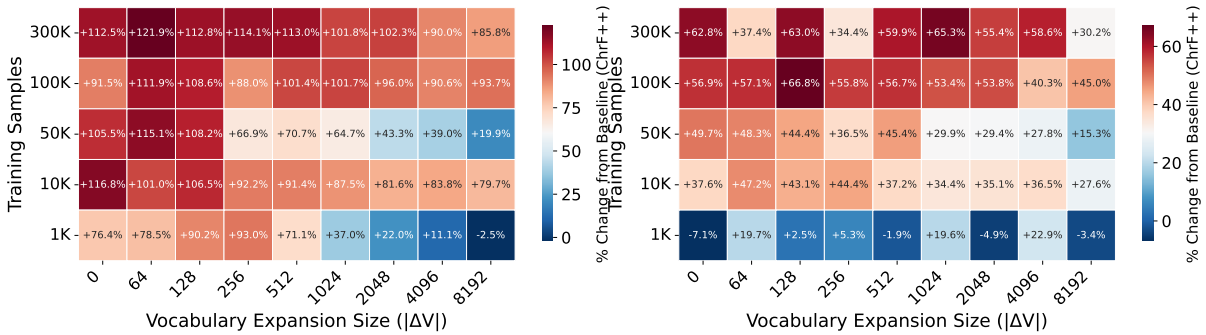


Figure 26: Gemma2-9B: MT Performance, English → Tibetan (left) / Sanskrit (right), % improvement over base (chrF++ 7.67 and 7.56 respectively).

Artifact	Type	License	Usage
CC-100[a]	Dataset	Not specified[**]	Data
Saamayik[b]	Dataset	MIT	Data
spsither Tibetan Corpus[c]	Dataset	Not specified[*]	Data
MC2 Corpus[d]	Dataset	CC0-1.0	Data
Tibetan–English Translation Dataset (Lotsawa House)[e]	Dataset	CC-BY-NC-4.0	Data
Language Model Evaluation Harness[f]	Python package	MIT	Evaluation
Claude-4.5-Sonnet[g]	API service	Commercial ToS	Evaluation

[a] https://data.statmt.org/cc-100/
[b] https://huggingface.co/datasets/acomquest/Saamayik
[c] https://huggingface.co/datasets/spsither/tibetan_monolingual_A_filtered_deduped
[d] https://huggingface.co/datasets/pkupie/mc2_corpus
[e] https://huggingface.co/datasets/billingsmoore/tibetan-to-english-translation-dataset
[f] https://github.com/EleutherAI/lm-evaluation-harness
[g] https://www.anthropic.com/legal/commercial-terms
[*] No license declared by the publisher on the dataset page; used solely for academic research.
[**] No license declared at the source; only a disclaimer over corpus preparation. Underlying Common Crawl text remains subject to its original copyright.

Table 9: Artifacts used during experiments, along with their license and usage explanation.