Parallel Tokenizers: Rethinking Encoder Models' Vocabulary Design in Cross-Lingual Transfer of Low-Resource Languages

Anonymous ACL submission

Abstract

Tokenization forms the basis of multilingual language models, yet existing methods often limit cross-lingual transfer by mapping semantically equivalent words to different embeddings. For example, "*I eat rice*" in English and "*Ina cin shinkafa*" in Hausa are typically mapped to different vocabulary indices, preventing shared representations and limiting cross-lingual generalization. This problem is even more pronounced in low-resource languages, where shared representations could offer the greatest benefit. We introduce **parallel tokenizers**, a new framework that first trains tokenizers monolingually and then aligns their vocabularies exhaustively using bilingual dictionaries or word-to-word translation. This alignment enforces a shared semantic space across languages while naturally improving fertility balance. To assess their effectiveness, we pretrain a transformer encoder from scratch on thirteen low-resource languages and evaluate it on sentiment analysis, hate speech detection, emotion classification, and sentence embedding similarity. Across all tasks, models trained with parallel tokenizers outperform conventional multilingual baselines, confirming that rethinking tokenization is essential for advancing multilingual representation learning–especially in low-resource settings.[1]

1 Introduction

Tokenization converts text into discrete units such as words, subwords, or bytes (Sennrich et al., 2016). In multilingual models (Devlin et al., 2019; Conneau et al., 2020a; Xue et al., 2021), a shared tokenizer is typically applied across languages, often disadvantaging underrepresented languages because the same semantic content is segmented into longer token sequences. This imbalance, measured by the fertility score (Rust et al., 2021), reveals systematic inefficiencies in multilingual tokenization. While shared embeddings across languages could improve representation learning, especially for low-resource languages, current tokenizers rarely achieve such alignment.

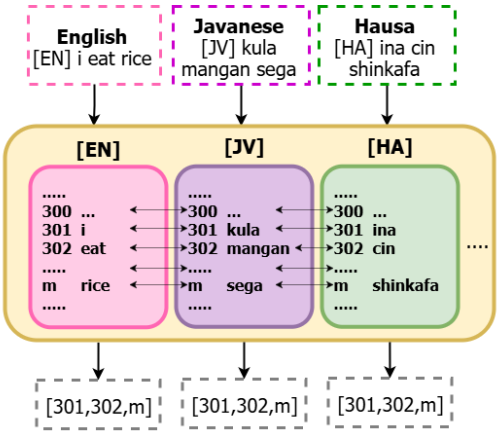

Figure 1: The overview of the parallel tokenizers. Tokens with equivalent meanings across languages are mapped to the same index and thus share the same embedding representation in the model.

Beyond fertility, multilingual tokenizers also fail to align semantics across languages. Semantically equivalent words (e.g., "*eat*" in English, "*ci*" in Hausa, and "食べる" in Japanese) are mapped to entirely distinct vocabulary indices (Devlin et al., 2019; Conneau et al., 2020a; Xue et al., 2021; Achiam et al., 2023; Bai et al., 2023; Dubey et al., 2024), erecting artificial barriers to cross-lingual sharing. Multilingual tokenizers also struggle with false cognates, where the same token may carry different meanings across languages. This fragmentation undermines the promise of transfer learning in multilingual models and introduces redundancy and inefficiency at the very foundation of representation learning.

Several strategies have been proposed to address these challenges, including approaches for pretraining medium- and low-resource language models (Koto et al., 2025) and byte-/character-level tok-

[1] Code can be found at anonymous.com

enizers (Xue et al., 2022; Clark et al., 2022; Cao, 2023). However, semantic redundancy remains unresolved, and reducing token length disparities often comes at the cost of efficiency and semantic interpretability

To strengthen shared representation across low-resource languages, we propose a **parallel tokenizer** (Figure 1). Instead of relying on a single multilingual tokenizer, we first train a monolingual tokenizer as a pivot and then align its vocabulary with target languages via word-level mapping, ensuring that token indices correspond to semantically equivalent words across languages. While achieving perfect alignment is inherently difficult, we restrict alignment to exact word forms. To reduce cross-lingual interference, we further incorporate language identity embeddings (Figure 2), ensuring that token representations remain unique to each language while still enabling cross-lingual semantic sharing. This design yields fairer representations, facilitates more effective transfer, and provides a scalable framework for adding new languages without retraining the entire vocabulary.

For evaluating our approach, we focus on encoder-only architectures because decoder-style generative models are considerably less suitable in low-resource settings: their generation quality degrades sharply when training data is limited. Encoder models, in contrast, emphasize semantic understanding and remain highly effective for applications such as classification and information retrieval. These properties make encoder-only architectures a clean and principled test bed for isolating the effects of our tokenization improvements, as also shown by prior works (Wang et al., 2022; Koto et al., 2024)

Our contributions can be summarized as follows:

- We propose a parallel tokenizer for multilingual language models that improves fertility balance across languages and strengthens cross-lingual transfer in encoder-only models, compared with traditional multilingual tokenizers.
- We demonstrate its effectiveness across multiple downstream tasks, including sentiment analysis, emotion classification, hate speech detection, and sentence embedding, all in low-resource settings.
- We analyze performance under varying data availability (1%, 10%, 50%, 100%), auxiliary languages, and evaluate fertility scores to measure tokenization efficiency.

2 Related Works

The standard practice in multilingual language modeling is to train a single tokenizer on a combined multilingual corpus (Achiam et al., 2023; Devlin et al., 2019; Xue et al., 2021; Dubey et al., 2024; Bai et al., 2023), typically using algorithms such as WordPiece (Devlin et al., 2019), Byte Pair Encoding (BPE) (Sennrich et al., 2016), the Unigram Language Model (Kudo, 2018), or SentencePiece (Kudo and Richardson, 2018). Despite their simplicity and scalability, these approaches face the well-documented *curse of multilinguality* (Conneau et al., 2020a): monolingual tokenizers consistently outperform shared multilingual ones in fertility and segmentation quality (Rust et al., 2021; Petrov et al., 2023), revealing a persistent trade-off between coverage and linguistic specialization. To mitigate these issues, prior studies have proposed balancing vocabulary allocation across languages (Zheng et al., 2021; Chung et al., 2020; Liang et al., 2023; Koto et al., 2025). However, while such methods improve fertility for low-resource languages, they often lead to redundant semantic representations, where tokens with equivalent meanings occupy distinct indices.

Beyond these engineering efforts, several studies have examined the conceptual role of tokenization in multilingual transfer. Conneau et al. (2020b) argue that parameter sharing is the primary factor enabling cross-lingual generalization. However, this line of research has not explored multilingual token embeddings in a parallel setting, where alignment across languages could better capture shared semantics. Limisiewicz et al. (2023) focus on lexical overlap, showing that surface-form similarity alone does not ensure semantic equivalence, leaving open the challenge of designing vocabularies that promote semantic alignment across languages.

Recent theoretical work further highlights the enduring importance of tokenization. Rajaraman et al. (2024) demonstrate that transformers trained on structured distributions struggle to learn effectively without discrete token boundaries, calling into question the feasibility of tokenization-free paradigms (Deiseroth et al., 2024). In this context, our work revisits multilingual tokenization from a new perspective, emphasizing that tokenization remains a foundational component for building semantically aligned and efficient multilingual language models.

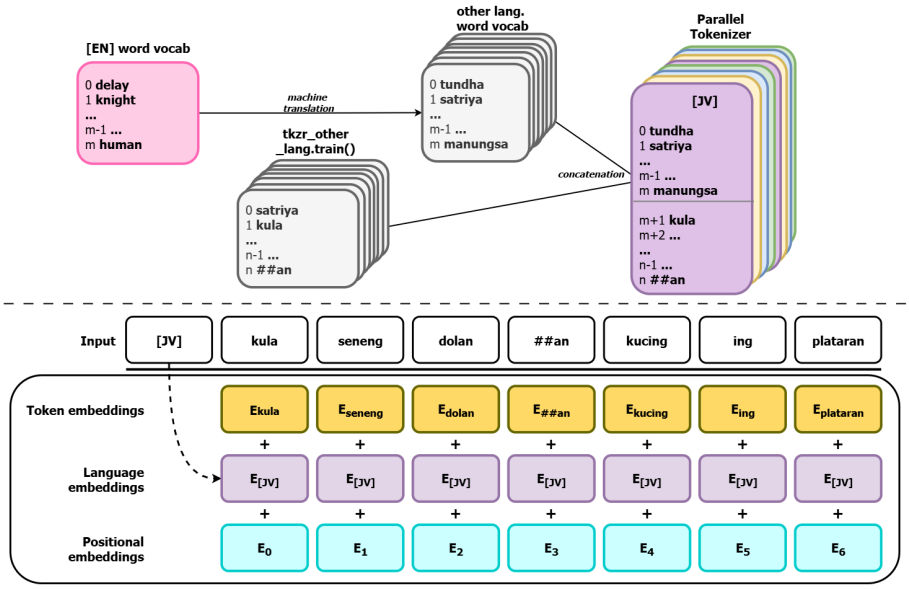

Figure 2: Design of the parallel vocabularies (top) and the model input representation (bottom). The language token (e.g., [JV] in the example) is not used as an explicit input token; instead, it functions as a signal to select the corresponding language identity embedding during input representation.

3 Parallel Tokenizer

To address fertility imbalance, false cognates, and the separation of semantically equivalent words into different token indices (Rust et al., 2021; Bai et al., 2023), we introduce the Parallel Tokenizer. This section outlines (1) the construction of the parallel vocabulary and (2) the design of the model input representation.

3.1 Parallel Vocabulary Design

To construct parallel vocabularies across languages, we first train an English monolingual tokenizer using WordPiece (Devlin et al., 2019) on the English Wikipedia dump as the corpus. We set the vocabulary size to 30,522. The resulting vocabulary entries can be grouped into four categories:

1. `subword`, e.g., *'##ing'*, *'##able'* (28.75% of vocabulary);
2. `short word`, e.g., *'ple'*, *'szo'*, *'rae'*; tokens with a length shorter than four characters (3.28% of vocabulary);
3. `number`, numerical tokens such as *'192'*, *'392'*, etc. (2.10% of vocabulary); and
4. `word`, which encompasses all other tokens, e.g., *'care'*, *'drink'*, *'administration'* (65.87% of vocabulary).

We retain only the English `word`-type vocabulary and expand it into other languages through token-level machine translation (see Figure 2). This filtering is necessary because other token types (`subword`, `short word`, and `number`) often yield unreliable or meaningless translations that rarely correspond to valid equivalents in the target languages. Using this filtered vocabulary as the base, we construct the parallel tokenizer in three steps: (1) translating the `word`-type vocabulary, (2) training a monolingual tokenizer for each target language, and (3) concatenating the resulting vocabularies.

Translating the `word`-type vocabulary. We translate each English `word`-type token into its counterpart in the target language using Google Translate[2]. We initially considered bilingual dictionaries such as PanLex[3], but found substantial limitations in both coverage and data quality, where only an average of 21.5% of the tokens are aligned. Machine translation offers broader and more up-to-date lexical coverage. To further ensure quality, we apply back-translation to the invalid outputs, such as multiword or malformed phrases. This process yields semantically aligned vocabularies that form the word-level backbone of our parallel tokenizer.

Training a monolingual tokenizer on the target language. Since the English tokenizer contains only 65.87% `word`-type tokens, we need to expand

[2] https://translate.google.com/
[3] https://old.panlex.org/

each parallel tokenizer to the full vocabulary size of 30,522 tokens by training a monolingual tokenizer on its corresponding Wikipedia corpus. The resulting vocabularies serve as sources for additional token types. We also introduce *language identity tokens* (e.g., [JV] in Figure 2) to explicitly mark the input language and ensure consistent handling across all languages (see Section 3.2).

Concatenating the vocabulary. For each target language, we concatenate two components: (1) the translated word-type tokens and (2) the corresponding monolingual vocabulary. We prioritize the translated word-type tokens, special tokens (e.g., [SEP], [UNK], [CLS])[4], and a 1,000-character-level subset from the English tokenizer. We then append the monolingual vocabulary, remove duplicates, and cap each final vocabulary at 30,522 entries for consistency across languages and fair comparison with the baseline. On average, across all languages defined in Section 4, we align 82% of word-type tokens (slightly below 100% due to back-translation filtering), and 61% of all tokens are aligned in the final parallel tokenizer.

3.2 Model Input Representation

Let $x = (l, x_1, x_2, \ldots, x_i)$ denote an input sequence of length i, where $l \in \mathcal{L}$ represents a language identity token and $\mathcal{L}$ is the set of all such tokens (e.g., [AM], [JV] in Figure 2), with $|\mathcal{L}| = k$. We define a collection of k parallel tokenizers, each corresponding to a specific language, as $\mathcal{T} = \{T_1, T_2, \ldots, T_k\}$.

Based on Figure 2, the language identity token l determines which tokenizer $T_j \in \mathcal{T}$ is used for tokenization. We then tokenize the remaining sequence $x' = (x_1, x_2, \ldots, x_i)$, excluding l, using T_j to obtain the token IDs $\mathcal{I} = (I_1, I_2, \ldots, I_m)$, where $i \leq m$, along with any auxiliary representations such as attention masks and token type IDs. The final input representation is constructed by summing the token embeddings, segment embeddings, positional embeddings, and a language identity embedding. The language identity embedding is broadcast across all m tokens to provide a unique representation for each language. This additional embedding serves both as a syntactic cue and as a disambiguation signal for unaligned tokens (recall from Section 3.1 that approximately 61% of tokens are aligned across languages).

[4]The complete list of language tokens is provided in Appendix A, Table 7, column 'code'.

4 Experimental Setup

4.1 Language Setup

We first define the set of languages considered in this work. English is included as the base language due to its large corpus size and its role in training the base tokenizer. To assess the generalization of our approach under limited-resource conditions, we additionally select a diverse set of low-resource languages. These include Javanese (jav), Minangkabau (min), Sundanese (sun), Swahili (swa), Acehnese (ace), Amharic (amh), Balinese (ban), Hausa (hau), Igbo (ibo), Kinyarwanda (kin), Oromo (orm), Tigrinya (tir), and Twi (twi), all of which are considered low-resource languages.

Most of these languages use the Latin script, while only Amharic and Tigrinya use the Amharic script. The selection is guided by the availability of essential resources, including Wikipedia dumps[5] for tokenizer and transformer pretraining, Google Translate for constructing parallel tokenizers, and FLORES+ (Costa-Jussà et al., 2022) for evaluating tokenization metrics.

4.2 Tokenizers & Models

To fairly compare tokenization strategies, we train two encoder-only models from scratch with identical pretraining setups but different tokenizers. The first uses a single shared tokenizer trained on the combined Wikipedia data of 13 languages, representing the standard multilingual approach; we refer to this setting as *Single-13L*. The second applies our proposed parallel tokenization method (*Parallel-13L*) trained on the same data. In both cases, model parameters are randomly initialized to isolate the effect of tokenization on multilingual representation learning.

4.3 Evaluation Setup

To evaluate the effectiveness of our proposed parallel tokenizer, we conduct three main experiments in low-resource languages: (i) tokenization analysis, (ii) sequence classification, and (iii) cross-lingual representation similarity.

Tokenization Analysis. We use the FLORES+ dataset (Costa-Jussà et al., 2022; Abdulmumin et al., 2024)[6], a parallel corpus covering 200 languages, to evaluate the effectiveness of our parallel

[5]https://dumps.wikimedia.org/. Resources were downloaded in November 2024.

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

tokenizer against both multilingual and monolingual tokenizers in 1,012 parallel texts. All 13 languages studied in our experiments are represented in FLORES+. We report two metrics: (i) Fertility score: the average number of tokens per word, indicating how compact or fragmented the tokenization is; and (ii) Parity score (Petrov et al., 2023): a cross-lingual metric that compares token counts across two languages,[7] assigning the best score when the counts are identical.

Sequence Classification. We conduct experiments on sentiment analysis, hate speech detection, and emotion classification in low-resource languages by fine-tuning the language model jointly on data from all languages. To analyze cross-lingual transfer, we vary the training size at 1%, 10%, 50%, and 100% of the available data. For sentiment analysis and hate speech detection, we use the NusaX-Senti (Winata et al., 2023) and AfriHate (Muhammad et al., 2025a) datasets, respectively, and evaluate both using macro-F1 over three multi-classes. For emotion classification, we use EthioEmo (Belay et al., 2025) and BRIGHTER (Muhammad et al., 2025b), evaluated with weighted F1 for consistency across six multi-label tasks. A detailed overview of the dataset languages is provided in Table 8 (Appendix B).

Cross-Lingual Representation Similarity. We perform bitext mining (Artetxe and Schwenk, 2019) on FLORES+ using the contextualized representations from the final layer of the language models. Following the setup of Huang et al. (2025), we evaluate **all possible language pairs** covered by our models. Performance is measured using the error rate of the `xsim` score (Artetxe and Schwenk, 2019). Additionally, we apply Principal Component Analysis (PCA) (Jolliffe, 2002) to visualize cross-lingual clustering, comparing our parallel tokenizer with the multilingual model that uses a single tokenizer. Representations are considered semantically aligned when sentences from different languages cluster together, rather than being separated by language identity.

4.4 Hyperparameters and Resources

We use an NVIDIA RTX A6000 GPU with 48GB of VRAM for both pretraining and fine-tuning. Pretraining follows the masked language modeling (MLM) setup using a learning rate of 5e-5 for the *Single-13L* model and 1e-4 for our proposed *Parallel-13L* model. Training is performed on 394M tokens per epoch (6M tokens for the development set) for a total of 122,850 steps or 50 epochs. The value is determined empirically based on the convergence of the pretraining loss on the development set.

For evaluation, we set the batch size to 20, the maximum sequence length to 128, and the maximum number of epochs to 100, with early stopping triggered after 5, 2, 3, and 3 epochs for NusaX-senti, AfriHate, EthioEmo, and BRIGHTER, respectively. Learning rates were adopted from the original benchmark recommendations: 1e-5 for NusaX-senti and BRIGHTER, and 5e-5 for AfriHate and EthioEmo. Each experiment was repeated three times with different seeds (1, 12, and 123), and we report the mean performance together with the corresponding standard deviations.

5 Results and Analysis

5.1 Tokenization Qualities

Table 1 compares our *Parallel-13L* tokenizer with *Single-13L* and the monolingual tokenizers. In terms of fertility score, our method substantially and consistently outperforms the multilingual baseline, exceeding by an average of 0.32 points, indicating more compact segmentation (i.e., fewer tokens per word). The only slight gap (0.05) appears against the monolingual tokenizer, which is expected since monolingual models are optimized for their respective languages. This highlights the effectiveness of our approach in reducing the number of tokens during multilingual tokenization, thereby lowering computational requirements.

For the parity score, which quantifies consistency in tokenization across parallel texts and reflects cross-lingual alignment, *Parallel-13L* achieves the best overall performance, outperforming the baseline (*Single-13L*) by 0.07 points on average. The *Single-13L* performs notably worse on Amharic-script languages (Amharic and Tigrinya) due to script-specific segmentation differences, while monolingual tokenizers score lowest by design, as parity inherently measures multilingual generalization.

5.2 Sequence Classification

Across three seeds, Table 2 shows that our parallel tokenizer (*Parallel-13L*) consistently outperforms the baseline across training sizes and evaluation

[7] We select Hausa as the reference language for comparison, as it provides the largest available resource during pretraining.

lang.	tokens/word (fertility score) ↓			parity score ↓		
	Multilingual Tokenizer		Mono-lingual	Multilingual Tokenizer		Mono-lingual
	Single-13L	Parallel-13L (ours)		Single-13L	Parallel-13L (ours)	
jav	1.65	**1.48**	1.43	**1.06**	1.13	1.69
min	1.79	**1.50**	1.46	**1.05**	1.09	1.68
sun	1.74	**1.50**	1.44	**1.01**	1.13	1.74
swa	1.65	**1.42**	1.38	**1.01**	1.10	1.63
ace	1.95	**1.54**	1.61	1.20	1.02	1.51
amh	2.43	**1.82**	1.66	1.21	**1.06**	2.33
ban	1.77	**1.53**	1.48	**1.03**	1.07	1.63
hau	1.38	**1.33**	1.30	1.00	1.00	1.00
ibo	1.54	**1.50**	1.48	**1.10**	1.12	1.32
kin	2.10	**1.70**	1.63	1.21	**1.03**	1.53
orm	2.40	**1.82**	1.72	1.35	**1.07**	1.55
tir	2.61	**1.83**	1.85	1.45	**1.06**	2.14
twi	1.55	**1.38**	1.35	1.16	**1.07**	1.42
avg	1.89	**1.57 (↓0.32)**	1.52	1.14	**1.07 (↓0.07)**	1.63

Table 1: Fertility and parity scores for each tokenizer across all languages. The **bold** scores indicate the best performance in multilingual tokenizers. Monolingual tokenizers are also included to provide a comparative perspective between monolingual and multilingual tokenizers.

#training data	Multilingual Tokenizer	*Sentiment*	*Hate Speech*	*Emotion Classification*		Avg ↑
		NusaX-senti ↑	AfriHate ↑	EthioEmo ↑	BRIGHTER ↑	
100%	Single-13L	76.09 (±1.27)	69.61 (±1.19)	54.98 (±0.98)	48.26 (±0.94)	62.24
	Parallel-13L (ours)	**76.16 (±1.05)**	**69.80 (±1.20)**	**57.01 (±0.84)**	**49.68 (±1.89)**	**63.16**
50%	Single-13L	72.47 (±1.36)	67.26 (±1.36)	51.33 (±1.48)	45.77 (±1.46)	59.21
	Parallel-13L (ours)	**73.52 (±1.17)**	**67.40 (±0.94)**	**54.27 (±1.78)**	**46.76 (±1.51)**	**60.49**
10%	Single-13L	64.76 (±1.84)	59.80 (±2.09)	**42.10 (±2.19)**	36.64 (±2.23)	50.83
	Parallel-13L (ours)	**66.16 (±2.00)**	**60.17 (±1.50)**	41.35 (±2.21)	**38.51 (±1.27)**	**51.55**
1%	Single-13L	31.54 (±3.87)	44.70 (±4.45)	**27.37 (±1.88)**	17.54 (±2.57)	30.29
	Parallel-13L (ours)	**33.83 (±3.90)**	**46.73 (±2.52)**	26.58 (±1.37)	**18.88 (±3.96)**	**31.51**

Table 2: Performance comparison of tokenizer setups on sentiment analysis, hate speech detection, and emotion classification tasks. Best scores per benchmark setups are highlighted in **bold**.

metrics. On average, *Parallel-13L* exceeds the baseline by 0.92%, 1.28%, 0.72%, and 1.22% F1 at the 100%, 50%, 10%, and 1% training data levels, respectively. These results demonstrate the effectiveness of a parallelized tokenizer design in enhancing cross-lingual transfer, particularly in low-resource scenarios.

The improvements in our approach are consistent across the NusaX-senti, AfriHate, and BRIGHTER datasets, where *Parallel-13L* outperforms the *Single-13L* baseline regardless of training size. The only exception occurs with EthioEmo, where *Single-13L* achieves higher scores than *Parallel-13L* only at the 10% and 1% settings. Detailed per-language benchmark results are provided in Appendix C (Tables 9, 10, 11, and 12 for NusaX-senti, AfriHate, EthioEmo, and BRIGHTER, respectively).

5.3 Cross-Lingual Representation Similarity

Using the parallel FLORES+ corpus, we visualize cross-lingual representation similarity for each model by extracting the final hidden states of 250 sampled parallel sentences. If a model learns shared representations across languages, PCA should reveal clusters based on semantic similarity rather than language identity. Figure 3 compares *Single-13L* and *Parallel-13L* in 2D space. *Single-13L* tends to cluster representations by language family, for example, Indonesian languages (Sundanese, Minangkabau, Balinese, and Javanese) group together, while African languages form a separate cluster. In contrast, *Parallel-13L* yields more compact cross-lingual clusters, indicating stronger semantic alignment, with only Acehnese, Oromo, and Tigrinya appearing as outliers for both due to limited data.

We further evaluate cross-lingual similarity through bitext mining. As shown in Table 3, *Parallel-13L* achieves the lowest `xsim` error rate and the highest number of best scores across language pairs (details in Appendix D). These results confirm that *Parallel-13L* learns more semantically consistent representations across languages than *Single-13L*, reinforcing its advantage for cross-lingual tasks.

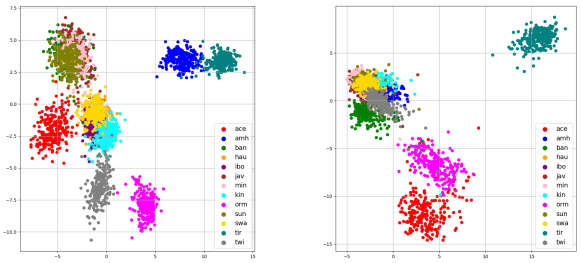

Figure 3: PCA visualization of the last hidden states from *Single-13L* (left) and *Parallel-13L* (right) models on the FLORES+ dataset.

	Single-13L	Parallel-13L (ours)
Avg. err. `xsim` ↓	83.56	**74.08**
# of best `xsim` ↑	12	**66**

Table 3: Bitext mining metric meta-scores on pretrained models under different setups.

#data in tgt language		Single-13L	Parallel-13L (ours)
50%	NusaX-senti	75.22	**75.42**
	AfriHate	**67.06**	67.00
	EthioEmo	51.73	**53.63**
	BRIGHTER	45.00	**47.11**
0%	NusaX-senti	71.83	**72.70**
	AfriHate	37.27	**40.67**
	EthioEmo	**23.75**	23.20
	BRIGHTER	15.04	**19.73**

Table 4: Benchmarking results under limited target-language data.

	mBERT	S-13L*	P-13L* (ours)
Sequence Classification Benchmark			
NusaX-senti	73.1 (±1.7)	78.3 (±1.0)	**78.6** (±1.2)
Afrihate	62.8 (±2.4)	**71.3** (±1.9)	70.1 (±0.9)
EthioEmo	30.1 (±7.0)	56.7 (±0.9)	**57.4** (±0.8)
BRIGHTER	48.5 (±1.3)	**50.5** (±0.9)	50.3 (±1.2)
Bitext Mining Benchmark			
Avg. `xsim` ↓	91.33	72.18	**69.34**
# of best `xsim` ↑	1	29	**48**

Table 5: Benchmarking results under the continual pretraining setting (top: sequence classification benchmark; bottom: bitext mining benchmark). S-13L* and P-13L* denote *Single-13L* and *Parallel-13L*, respectively, both of which are continually pretrained from the mBERT model.

5.4 Cross-Lingual Transfer under Limited Target-Language Data

While our main experiments utilize data from all languages during training, we also analyze the model's behavior when the target-language data is scarce or unavailable–a common scenario in low-resource settings. To simulate this condition, we exclude the target language from the training corpus and evaluate two setups: (i) 0% target-language data (zero-shot) and (ii) 50% target-language data (partial-shot). Table 4 shows that our method consistently achieves the highest F1 scores in both settings. The only exceptions are a slight decrease of 0.06% on AfriHate with 50% target data and 0.55% on EthioEmo with 0% target train data. Overall, *Parallel-13L* outperforms the baseline, demonstrating stronger cross-lingual transfer than the traditional single-shared tokenizer. The detailed results are presented in Appendix E in Table 15, 16, 17, and 18.

5.5 Continual Pre-Training Experiments

We perform continual pretraining (CPT) from mBERT (Devlin et al., 2019) by adapting its vocabulary and tokenizer to our parallel setup. Newly introduced token embeddings are initialized by averaging the mBERT embeddings of their constituent subwords, as determined by the original mBERT tokenizer (Koto et al., 2021). Unlike pretraining from scratch, this approach retains all mBERT parameters while updating the token embedding layer to accommodate the new vocabulary. In *Single-13L*, each token in the new vocabulary is mapped to its corresponding English subwords in mBERT, and its embedding is initialized by averaging the associated subword embeddings. In *Parallel-13L*, each token is tokenized into subwords across all 13 languages, and its embedding is initialized by averaging the pooled subword embeddings across languages.

Table 5 shows that *Single-13L* and *Parallel-13L* achieve similar sequence classification performance under continual pretraining (CPT), and both outperform the original mBERT when it is fine-tuned directly without CPT. However, our method yields stronger cross-lingual alignment (see Figure 5 in Appendix G), and bitext mining benchmarking confirms more consistent representations than the baselines. This indicates that even when task scores converge, *Parallel-13L* maintains a structural advantage in preserving cross-lingual coherence.

5.6 Unseen Languages Performance in Downstream Task

We further analyze the role of language identity within the parallel tokenizer framework, particularly for unseen languages. This evaluation is motivated by practical scenarios where the input language may not be included in the multilingual training set, or when an incorrect language iden-

#training	tokenizer	bbc	bjn	bug	mad	nij	avg
100%	S-13L	**52.87**	**69.00**	**51.08**	61.76	59.92	**58.93**
	P-13L (Best)	51.67	65.50	48.80	**61.81**	**60.72**	57.70
	P-13L (Geo)	51.67	65.50	40.26	59.67	53.71	54.16
	P-13L (Avg)	46.73	62.13	44.29	60.64	54.62	53.68
1%	S-13L	26.70	28.72	29.10	28.38	30.36	28.65
	P-13L (Best)	**39.37**	**36.63**	**38.91**	**35.54**	**38.37**	**37.76**
	P-13L (Geo)	31.56	34.06	26.55	33.59	30.26	31.20
	P-13L (Avg)	34.73	32.59	30.13	33.72	31.69	32.57

Table 6: Benchmarking results for unseen languages in NusaX-senti. S-13L and P-13L denote *Single-13L* and *Parallel-13L*, respectively. For the parallel setting, 'Best' denotes the best-performing tokenizer pair, 'Geo' selects the tokenizer based on the geographically closest language, and 'Avg' reports the average performance obtained using all tokenizers for that language.

tity is inadvertently supplied at inference time. In such cases, the parallel tokenizer is forced to operate with a mismatched tokenizer-language identity pairing, allowing us to assess how robustly the model handles mis-specified or unseen language contexts.

To examine this effect, we design an extreme evaluation scenario using **five unseen languages** from the NusaX-senti dataset: bbc (Toba Batak), bjn (Banjarnese), bug (Buginese), mad (Maduranese), and nij (Ngaju). It is worth noting that this scenario is rare in practice, since the target language is typically known when applying multilingual models. Moreover, such a scenario is even less likely given that language identification systems are highly accurate (Adebara et al., 2022). We evaluate *Single-13L* and *Parallel-13L* under two training data regimes: 100% and 1% of the available training data.

Table 6 shows that when *Parallel-13L* is forced to use an incorrect language identity, *Single-13L* performs better in the full-data (100%) setting. However, in the 1% setting, which better reflects realistic unseen and/or low-resource language scenarios, the parallel tokenizer demonstrates clear advantages, even when the tokenizer is selected based on geographical proximity (Geo) or averaged across all tokenizers (Avg). This highlights a trade-off between stability in high-resource settings and robustness in low-resource scenarios; while such extreme mismatches are rare in practice, our method shows superior generalization when training data is severely limited.

6 Discussion

As described in Section 3.1, we use machine translation (MT) via Google Translate API to translate English vocabulary into several low-resource languages. Because the system is primarily designed for sentence-level translation, we evaluate its word-level performance to ensure proper alignment with the English vocabulary. To examine that, we sample 100 vocabulary items each from Minangkabau (`min`), Javanese (`jav`), and Sundanese (`sun`), and ask fluent annotators to assess and mark whether the translations are correct or not. The average accuracy is 83% (details: `min`=76%, `jav`=84%, `sun`=89%). While the alignment is not perfect, the MT-based approach perceives substantial scores, given that the languages themselves are low-resource.

Nevertheless, even when the parallel tokenizer produces multiple subword units instead of a single full-form token, our approach still offers clear advantages over traditional multilingual tokenizers. Because we attach a language embedding to every token, each subword retains a language-specific identity, eliminating cross-lingual semantic ambiguity. In contrast, standard multilingual tokenizers reuse identical subword pieces across languages, regardless of whether they share meaning, often conflating unrelated semantics. Importantly, our fertility scores empirically confirm that the proposed tokenizer yields more compact and semantically faithful segmentations, and these improvements translate into consistent performance gains across downstream tasks.

7 Conclusion

We introduce parallel tokenizers, constructed by first training a monolingual tokenizer as a pivot and then aligning its word vocabulary with other languages using machine translation, ensuring that semantically equivalent words across languages share the same representation. We evaluate this approach along three dimensions. In terms of tokenization quality, it yields the lowest fertility and parity scores, indicating more efficient tokenization while reducing disparities across languages. In sequence classification, our method shows improvements over the baselines, yielding higher average F1 scores across benchmarks. Finally, in cross-lingual sentence representation, our approach produces stronger cross-lingual alignment representation than ordinary tokenization methods, making it especially beneficial for cross-lingual training and for incorporating auxiliary languages into encoder-only multilingual models.

Limitations

To develop a robust parallel tokenizer, each stage of designing the parallel vocabulary and model input representation required careful consideration. For mapping the English vocabulary in the tokenizer, we employed machine translation (MT). Although bilingual dictionaries can potentially provide more precise word-level mappings due to their predefined vocabularies, our attempt to utilize the PanLex bilingual dictionary (as discussed in Section 3.1) revealed substantial limitations in both coverage and data quality, particularly for lower-resource languages. While MT enabled us to align approximately 61% of tokens, the bilingual dictionary achieved only about 21.5%. Therefore, we adopted MT for our final approach.

This choice, however, introduces a minor limitation related to the quality of the machine translation itself, as several mapping errors were observed, including invalid outputs such as multiword or malformed phrases. Consequently, there remains potential to improve token alignment through more refined resources or hybrid methods. Furthermore, there is room to enhance the model's input representation, as detailed in Appendix H, which we plan to investigate further in future work.

Our current study focuses on sequence classification, aligning with our focus on low-resource languages that have limited data. In future research, as more diverse datasets become available, we aim to expand the benchmarking to additional tasks and languages, ensuring that the parallel tokenizer can effectively scale and maintain performance across a broader linguistic spectrum.

Ethical Consideration

All datasets used in this work are publicly available and comply with their respective licenses, including Wikipedia, FLORES+, and established benchmarks for sentiment, hate speech, and emotion classification. No personally identifiable or sensitive information was collected or used. Our study aims to improve multilingual representation quality, particularly for low-resource languages. However, since the underlying data may contain cultural or societal biases, we recommend that future work include bias analysis and community engagement when extending models to additional linguistic or cultural contexts.

References

Idris Abdulmumin, Sthembiso Mkhwanazi, Mahlatse S. Mbooi, Shamsuddeen Hassan Muhammad, Ibrahim Said Ahmad, Neo N. Putini, Miehleketo Mathebula, Matimba Shingange, Tajuddeen Gwadabe, and Vukosi Marivate. 2024. Correcting FLORES evaluation dataset for four African languages. In *Proceedings of the Ninth Conference on Machine Translation*, Miami, USA. Association for Computational Linguistics.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Ife Adebara, AbdelRahim Elmadany, Muhammad Abdul-Mageed, and Alcides Inciarte. 2022. AfroLID: A neural language identification tool for African languages. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1958–1981, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Mikel Artetxe and Holger Schwenk. 2019. Margin-based parallel corpus mining with multilingual sentence embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3197–3203, Florence, Italy. Association for Computational Linguistics.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, and 1 others. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Tadesse Destaw Belay, Israel Abebe Azime, Abinew Ali Ayele, Grigori Sidorov, Dietrich Klakow, Philip Slusallek, Olga Kolesnikova, and Seid Muhie Yimam. 2025. Evaluating the capabilities of large language models for multi-label emotion understanding. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 3523–3540, Abu Dhabi, UAE. Association for Computational Linguistics.

Kris Cao. 2023. What is the best recipe for character-level encoder-only modelling? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5924–5938, Toronto, Canada. Association for Computational Linguistics.

Hyung Won Chung, Dan Garrette, Kiat Chuan Tan, and Jason Riesa. 2020. Improving multilingual models with language-clustered vocabularies. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4536–4546, Online. Association for Computational Linguistics.

Jonathan H. Clark, Dan Garrette, Iulia Turc, and John Wieting. 2022. Canine: Pre-training an efficient

tokenization-free encoder for language representation. *Transactions of the Association for Computational Linguistics*, 10:73–91.

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

Alexis Conneau, Shijie Wu, Haoran Li, Luke Zettlemoyer, and Veselin Stoyanov. 2020b. Emerging cross-lingual structure in pretrained language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6022–6034, Online. Association for Computational Linguistics.

Marta R Costa-Jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, and 1 others. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.

Björn Deiseroth, Manuel Brack, Patrick Schramowski, Kristian Kersting, and Samuel Weinbach. 2024. T-FREE: Subword tokenizer-free generative LLMs via sparse representations for memory-efficient embeddings. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 21829–21851, Miami, Florida, USA. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407.

Yongxin Huang, Kexin Wang, Goran Glavaš, and Iryna Gurevych. 2025. Modular sentence encoders: Separating language specialization from cross-lingual alignment. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2167–2187, Vienna, Austria. Association for Computational Linguistics.

I. T. Jolliffe. 2002. *Principal Component Analysis*, 2nd edition. Springer Series in Statistics. Springer New York, NY, New York.

Fajri Koto, Tilman Beck, Zeerak Talat, Iryna Gurevych, and Timothy Baldwin. 2024. Zero-shot sentiment analysis in low-resource languages using a multilingual sentiment lexicon. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 298–320, St. Julian's, Malta. Association for Computational Linguistics.

Fajri Koto, Rituraj Joshi, Nurdaulet Mukhituly, Yuxia Wang, Zhuohan Xie, Rahul Pal, Daniil Orel, Parvez Mullah, Diana Turmakhan, Maiya Goloburda, and 1 others. 2025. Sherkala-chat: Building a state-of-the-art llm for kazakh in a moderately resourced setting. In *Second Conference on Language Modeling*.

Fajri Koto, Jey Han Lau, and Timothy Baldwin. 2021. IndoBERTweet: A pretrained language model for Indonesian Twitter with effective domain-specific vocabulary initialization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10660–10668, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia. Association for Computational Linguistics.

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.

Davis Liang, Hila Gonen, Yuning Mao, Rui Hou, Naman Goyal, Marjan Ghazvininejad, Luke Zettlemoyer, and Madian Khabsa. 2023. XLM-V: Overcoming the vocabulary bottleneck in multilingual masked language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13142–13152, Singapore. Association for Computational Linguistics.

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

Shamsuddeen Hassan Muhammad, Idris Abdulmumin, Abinew Ali Ayele, David Ifeoluwa Adelani, Ibrahim Said Ahmad, Saminu Mohammad Aliyu, Paul Röttger, Abigail Oppong, Andiswa Bukula, Chiamaka Ijeoma Chukwuneke, Ebrahim Chekol Jibril, Elyas Abdi Ismail, Esubalew Alemneh, Hagos Tesfahun Gebremichael, Lukman Jibril Aliyu,

Meriem Beloucif, Oumaima Hourrane, Rooweither Mabuya, Salomey Osei, and 8 others. 2025a. Afri-Hate: A multilingual collection of hate speech and abusive language datasets for African languages. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1854–1871, Albuquerque, New Mexico. Association for Computational Linguistics.

Shamsuddeen Hassan Muhammad, Nedjma Ousidhoum, Idris Abdulmumin, Jan Philip Wahle, Terry Ruas, Meriem Beloucif, Christine de Kock, Nirmal Surange, Daniela Teodorescu, Ibrahim Said Ahmad, David Ifeoluwa Adelani, Alham Fikri Aji, Felermino D. M. A. Ali, Ilseyar Alimova, Vladimir Araujo, Nikolay Babakov, Naomi Baes, Ana-Maria Bucur, Andiswa Bukula, and 29 others. 2025b. BRIGHTER: BRIdging the gap in human-annotated textual emotion recognition datasets for 28 languages. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8895–8916, Vienna, Austria. Association for Computational Linguistics.

Aleksandar Petrov, Emanuele La Malfa, Philip Torr, and Adel Bibi. 2023. Language model tokenizers introduce unfairness between languages. In *Advances in Neural Information Processing Systems*, volume 36, pages 36963–36990. Curran Associates, Inc.

Nived Rajaraman, Jiantao Jiao, and Kannan Ramchandran. 2024. An analysis of tokenization: Transformers under markov data. In *Advances in Neural Information Processing Systems*, volume 37, pages 62503–62556. Curran Associates, Inc.

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.

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.

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

Genta Indra Winata, Alham Fikri Aji, Samuel Cahyawijaya, Rahmad Mahendra, Fajri Koto, Ade Romadhony, Kemal Kurniawan, David Moeljadi, Radityo Eko Prasojo, Pascale Fung, Timothy Baldwin, Jey Han Lau, Rico Sennrich, and Sebastian Ruder. 2023. NusaX: Multilingual parallel sentiment dataset for 10 Indonesian local languages. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 815–834, Dubrovnik, Croatia. Association for Computational Linguistics.

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

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Bo Zheng, Li Dong, Shaohan Huang, Saksham Singhal, Wanxiang Che, Ting Liu, Xia Song, and Furu Wei. 2021. Allocating large vocabulary capacity for cross-lingual language model pre-training. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3203–3215, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

A Language Resource Availability Details

Table 7 presents the availability of language resources under study in the Wikipedia dumps. English is highlighted in lime, as it serves both as the reference for comparison to other lower-resource languages and the pivot language when constructing the parallel tokenizers, due to its large corpus size and richer vocabulary coverage.

Lang.	iso	code	Script	Resource
English	eng	[EN]	Latin	10.7GB
Acehnese	ace	[AC]	Latin	2.52MB
Amharic	amh	[AM]	Amharic	18MB
Balinese	ban	[BA]	Latin	14.24MB
Hausa	hau	[HA]	Latin	108MB
Igbo	ibo	[IG]	Latin	104MB
Javanese	jav	[JV]	Latin	50.7MB
Kinyarwanda	kin	[RW]	Latin	12.25MB
Minangkabau	min	[MI]	Latin	85.4MB
Oromo	orm	[OR]	Latin	3.71MB
Sundanese	sun	[SU]	Latin	34.1MB
Swahili	swa	[SW]	Latin	54.4MB
Tigrinya	tir	[TI]	Amharic	1.1MB
Twi	twi	[TW]	Latin	8.6MB

Table 7: Language resource availability. The 'iso' column lists language codes following the ISO 639-3 convention, while the 'code' column specifies the language identifiers used to determine the language embeddings in the model's input representation.

B Benchmark Language Coverage

Table 8 reports the language coverage of each benchmark included in our study. For brevity, we use the abbreviations NusaX, Afri, Ethio, and BRIGHT to refer to the NusaX-senti, AfriHate, EthioEmo, and BRIGHTER datasets, respectively. The numbers next to the checkmarks indicate the number of instances used in the benchmarking process.

lang.	NusaX	Afri	Ethio	BRIGHT
ace	✓(1K)			
amh		✓(4.96K)	✓(5.92K)	
ban	✓(1K)			
hau		✓(6.64K)		✓(5.02K)
ibo		✓(5K)		✓(6.73K)
jav	✓(1K)			
min	✓(1K)			
kin		✓(4.72K)		✓(5.73K)
orm		✓(5.03K)	✓(5.74K)	
sun	✓(1K)			✓(3.17K)
swa		✓(21.1K)		✓(7.72K)
tir		✓(5.07K)	✓(6.14K)	
twi		✓(3.9K)		

Table 8: Benchmark language coverage.

C Benchmarking Result Details

Tables 9, 10, 11, and 12 present the benchmarking results for NusaX-senti, AfriHate, EthioEmo, and BRIGHTER, respectively. Results are reported for each language in the benchmark under training data settings of 100%, 50%, 10%, and 1%. To ensure stability and reliability, each experiment was run three times with different random seeds, and the reported results include the corresponding standard deviations. Based on the average results, our method yields better results than the baseline, highlighting the performance of our method in cross-lingual settings.

D Bitext Mining Details

Tables 13 and 14 present the bitext mining performance of each model using their corresponding tokenizers: *Single-13L* and *Parallel-13L*. Performance is evaluated using the `xsim` error rate, which measures how accurately a model selects the equivalent sentence across languages based on the FLORES+ parallel data. A lower error rate indicates better cross-lingual alignment, making this metric an effective indicator of a model's ability to capture cross-lingual representations.

E Cross-Lingual Transfer under Limited Target-Language Data Details

Similar to Appendix C, we report benchmarking results under a limited target-language data setting, where the auxiliary languages are fully utilized (100% of the data) during finetuning, while the target language is limited to 50% or 0% of its data. This setup highlights cross-lingual transfer from the auxiliary languages to the target language. Tables 15, 16, 17, and 18 present the results for NusaX-senti, AfriHate, EthioEmo, and BRIGHTER, respectively. For each language, we repeat the experiments three times with different seeds to ensure robustness.

E.1 How Much Does Multilingual Training Improve Over Single-Language Training?

This analysis compares fine-tuning on concatenated multilingual datasets versus training solely on a target language's data. As shown in Figure 4, both baselines and our method benefit from multilingual training in the AfriHate benchmark, with average gains of 7-8%. In contrast, improvements are smaller for BRIGHTER and EthioEmo, while in NusaX-senti, our method still achieves a modest

#data	Tokenizer	ace	ban	jav	min	sun	**avg**
100%	Single-13L	**76.61** (±1.31)	**73.23** (±1.00)	77.43 (±1.45)	76.48 (±1.37)	76.68 (±1.19)	76.09 (±1.27)
	Parallel-13L (ours)	74.22 (±1.91)	72.62 (±1.22)	**78.26** (±0.50)	**78.43** (±0.42)	**77.26** (±1.22)	**76.16** (±1.05)
50%	Single-13L	**73.44** (±1.41)	69.68 (±1.76)	71.86 (±2.03)	73.14 (±1.01)	74.22 (±0.61)	72.47 (±1.36)
	Parallel-13L (ours)	71.53 (±0.52)	**72.83** (±2.03)	**74.34** (±0.86)	**74.38** (±2.15)	**74.52** (±0.31)	**73.52** (±1.17)
10%	Single-13L	63.63 (±2.35)	63.02 (±1.92)	**66.59** (±1.86)	65.06 (±1.60)	65.51 (±1.46)	64.76 (±1.84)
	Parallel-13L (ours)	**65.38** (±0.47)	**65.07** (±1.38)	65.65 (±5.20)	**68.62** (±1.19)	**66.07** (±1.77)	**66.16** (±2.00)
1%	Single-13L	**32.98** (±3.42)	30.46 (±5.47)	28.12 (±4.77)	34.90 (±2.78)	31.26 (±2.93)	31.54 (±3.87)
	Parallel-13L (ours)	24.78 (±6.28)	**33.14** (±2.55)	**39.18** (±6.41)	**37.75** (±2.91)	**34.29** (±1.34)	**33.83** (±3.90)

Table 9: NusaX-senti benchmarking results. 'ace', 'ban', 'jav', 'min', and 'sun' denote Acehnese, Balinese, Javanese, Minangkabau, and Sundanese, respectively.

#data	Tokenizer	amh	hau	ibo	kin	orm	swa	tir	twi	**avg**
100%	Single-13L	59.72 (±1.80)	69.51 (±1.94)	85.36 (±0.74)	72.96 (±1.54)	**62.21** (±0.73)	**90.29** (±0.39)	60.91 (±1.09)	**55.91** (±1.27)	69.61 (±1.19)
	Parallel-13L (ours)	**59.91** (±0.35)	**70.93** (±2.19)	**87.62** (±0.85)	**73.44** (±1.56)	61.45 (±0.59)	89.94 (±0.22)	**61.61** (±1.07)	53.47 (±2.77)	**69.80** (±1.20)
50%	Single-13L	55.15 (±1.98)	66.23 (±1.76)	84.30 (±0.20)	**71.21** (±1.57)	59.44 (±0.77)	87.41 (±0.46)	**61.33** (±1.21)	**53.01** (±2.95)	67.26 (±1.36)
	Parallel-13L (ours)	**57.23** (±0.65)	**67.89** (±2.15)	**85.47** (±0.67)	69.39 (±1.45)	**60.16** (±0.86)	**88.60** (±0.08)	59.50 (±0.76)	50.96 (±0.92)	**67.40** (±0.94)
10%	Single-13L	50.83 (±0.29)	61.34 (±0.87)	77.39 (±1.80)	56.89 (±1.52)	50.49 (±6.57)	82.78 (±1.13)	**51.52** (±2.53)	**47.13** (±1.98)	59.80 (±2.09)
	Parallel-13L (ours)	**51.27** (±2.07)	**62.50** (±1.78)	**79.93** (±1.09)	**63.93** (±0.15)	**51.08** (±0.38)	84.24 (±1.19)	49.10 (±1.47)	39.29 (±3.87)	**60.17** (±1.50)
1%	Single-13L	41.41 (±2.79)	**43.31** (±7.82)	47.25 (±6.60)	42.37 (±3.92)	41.98 (±4.11)	69.11 (±2.53)	**41.49** (±4.06)	30.70 (±3.79)	44.70 (±4.45)
	Parallel-13L (ours)	**42.70** (±1.03)	43.09 (±3.84)	**55.44** (±3.25)	**48.10** (±1.81)	42.35 (±3.00)	**71.33** (±2.22)	39.56 (±2.28)	**31.26** (±2.74)	**46.73** (±2.52)

Table 10: AfriHate benchmarking results. 'amh', 'hau', 'ibo', 'kin', 'orm', 'swh', 'tir', and 'twi' denote Amharic, Hausa, Igbo, Kinyarwanda, Oromo, Swahili, Tigrinya, and Twi, respectively.

#data	Tokenizer	amh	orm	tir	**avg**
100%	Single-13L	55.07 (±1.01)	58.96 (±1.18)	50.92 (±0.75)	54.98 (±0.98)
	Parallel-13L (ours)	**60.22** (±0.32)	59.59 (±1.18)	**51.21** (±1.03)	**57.01** (±0.84)
50%	Single-13L	52.98 (±2.09)	55.53 (±1.06)	45.48 (±1.31)	51.33 (±1.48)
	Parallel-13L (ours)	**58.33** (±1.12)	55.64 (±1.78)	**48.86** (±2.44)	**54.27** (±1.78)
10%	Single-13L	**44.32** (±2.86)	43.18 (±1.03)	**38.79** (±2.66)	**42.10** (±2.19)
	Parallel-13L (ours)	44.00 (±2.86)	**43.36** (±1.73)	36.68 (±2.04)	41.35 (±2.21)
1%	Single-13L	**34.85** (±0.34)	24.37 (±3.50)	**22.91** (±1.81)	**27.37** (±1.88)
	Parallel-13L (ours)	33.77 (±1.83)	**24.42** (±1.67)	21.55 (±0.62)	26.58 (±1.37)

Table 11: EthioEmo benchmarking results. 'amh', 'orm', and 'tir' denote Amharic, Oromo, and Tigrinya, respectively.

#data	Tokenizer	hau	ibo	kin	sun	swa	avg
100%	Single-13L	58.66 (±0.16)	**58.72** (±0.63)	41.65 (±0.58)	60.11 (±0.61)	22.16 (±2.71)	48.26 (±0.94)
	Parallel-13L (ours)	**60.20** (±1.96)	58.37 (±1.57)	**45.41** (±2.10)	60.49 (±1.05)	23.95 (±2.76)	**49.68** (±1.89)
50%	Single-13L	53.89 (±1.29)	**55.08** (±0.58)	39.36 (±1.83)	**59.77** (±1.05)	20.76 (±2.52)	45.77 (±1.46)
	Parallel-13L (ours)	**56.65** (±0.81)	55.05 (±1.65)	**43.42** (±0.81)	57.56 (±2.02)	21.13 (±2.26)	**46.76** (±1.51)
10%	Single-13L	43.03 (±1.65)	**45.12** (±2.80)	27.62 (±0.60)	50.63 (±4.83)	16.78 (±1.25)	36.64 (±2.23)
	Parallel-13L (ours)	**45.73** (±0.70)	45.00 (±2.27)	**33.12** (±0.68)	51.95 (±2.11)	16.74 (±0.58)	**38.51** (±1.27)
1%	Single-13L	9.02 (±8.00)	**21.28** (±3.23)	14.89 (±0.76)	42.50 (±0.84)	0.00 (±0.00)	17.54 (±2.57)
	Parallel-13L (ours)	**14.91** (±3.24)	15.97 (±13.91)	**17.20** (±1.93)	**46.30** (±0.72)	0.00 (±0.00)	**18.88** (±3.96)

Table 12: BRIGHTER benchmarking results. 'hau', 'ibo', 'kin', 'sun', and 'swa' denote Hausa, Igbo, Kinyarwanda, Sundanese, and Swahili, respectively.

	ace	amh	ban	hau	ibo	jav	min	kin	orm	sun	swa	tir	twi
ace		**97.83**	82.21	85.67	88.34	73.12	**72.83**	**91.01**	97.04	**77.57**	86.26	**98.52**	**89.62**
amh			96.74	96.34	96.94	96.44	96.74	97.53	97.73	96.94	96.54	90.51	98.02
ban				72.83	77.87	**35.97**	48.02	84.19	96.34	47.43	71.54	98.22	85.28
hau					64.43	62.15	72.13	82.91	95.26	68.38	63.54	97.63	81.72
ibo						70.95	78.46	87.15	96.84	75.69	71.05	98.02	85.97
jav							43.58	85.38	96.44	**34.88**	63.64	98.42	83.70
min								84.19	96.34	**42.89**	69.76	98.32	84.09
kin									97.04	**84.68**	78.36	98.32	88.54
orm										97.23	96.64	**98.12**	**96.84**
sun											67.29	98.62	84.58
swa												98.52	82.11
tir													98.91
twi													

Table 13: *Single-13L* bitext mining scores, calculated using `xsim` error rate score.

	ace	amh	ban	hau	ibo	jav	min	kin	orm	sun	swa	tir	twi
ace		98.32	88.04	**76.98**	**85.77**	**71.34**	81.23	98.62	97.92	82.31	**70.36**	98.81	98.22
amh			**77.87**	**73.62**	**85.08**	**69.96**	**72.43**	**83.30**	**95.45**	**81.92**	**65.61**	97.33	**86.96**
ban				**57.11**	**68.18**	50.79	**46.25**	**70.95**	**85.87**	**46.25**	**47.73**	**95.45**	**76.48**
hau					**52.87**	**35.57**	**41.40**	**60.18**	**82.91**	**57.11**	**30.93**	**89.43**	**70.55**
ibo						**58.10**	**68.28**	**78.36**	**87.35**	**72.53**	**54.25**	**97.23**	**77.27**
jav							**22.92**	**56.92**	**79.84**	44.27	**24.51**	**92.98**	**75.30**
min								**60.77**	**79.45**	87.15	**29.74**	**95.16**	**71.44**
kin									**92.29**	89.03	**59.19**	**96.15**	**81.92**
orm										**93.08**	**83.70**	98.62	97.23
sun											**47.33**	**92.89**	**72.92**
swa												**87.65**	**70.36**
tir													**98.62**
twi													

Table 14: *Parallel-13L* bitext mining scores, calculated using `xsim` error rate score.

#data	Tokenizer	ace	ban	jav	min	sun	avg
50%	Single-13L	**75.99** (±0.88)	70.69 (±2.97)	**78.00** (±0.82)	74.92 (±2.56)	**76.50** (±0.72)	75.22 (±1.59)
	Parallel-13L (ours)	73.75 (±0.62)	**71.94** (±1.05)	77.01 (±1.52)	**78.15** (±0.70)	76.22 (±2.10)	**75.42** (±1.20)
0%	Single-13L	66.77 (±2.59)	**69.76** (±2.20)	73.93 (±2.37)	74.09 (±1.85)	**74.60** (±0.95)	71.83 (±1.99)
	Parallel-13L (ours)	**69.44** (±1.94)	67.43 (±1.27)	**76.66** (±0.71)	**75.87** (±2.05)	74.12 (±1.50)	**72.70** (±1.49)

Table 15: NusaX-senti benchmarking results under a limited target-language data setting. 'ace', 'ban', 'jav', 'min', and 'sun' denote Acehnese, Balinese, Javanese, Minangkabau, and Sundanese, respectively.

#data	Tokenizer	amh	hau	ibo	kin	orm	swh	tir	twi	avg
50%	Single-13L	**57.92** (±0.69)	65.35 (±1.71)	84.20 (±0.43)	59.20 (±1.52)	69.23 (±0.80)	88.22 (±0.55)	57.75 (±3.59)	54.63 (±2.30)	**67.06** (±1.45)
	Parallel-13L (ours)	57.01 (±0.13)	66.18 (±1.73)	**85.27** (±1.52)	**59.35** (±2.04)	**70.77** (±0.62)	**88.73** (±0.11)	**58.57** (±1.41)	50.13 (±3.01)	67.00 (±1.32)
0%	Single-13L	42.32 (±2.74)	39.32 (±3.11)	38.91 (±1.91)	36.86 (±0.78)	40.06 (±4.86)	42.06 (±0.67)	35.27 (±2.89)	23.37 (±4.55)	37.27 (±2.69)
	Parallel-13L (ours)	**46.38** (±2.74)	**45.13** (±3.04)	**45.13** (±4.18)	**41.32** (±1.07)	**44.78** (±4.19)	**44.11** (±5.10)	32.45 (±2.41)	**26.08** (±1.68)	**40.67** (±3.05)

Table 16: AfriHate benchmarking results under a limited target-language data setting. 'amh', 'hau', 'ibo', 'kin', 'orm', 'swh', 'tir', and 'twi' denote Amharic, Hausa, Igbo, Kinyarwanda, Oromo, Swahili, Tigrinya, and Twi, respectively.

#data	Tokenizer	amh	orm	tir	avg
50%	Single-13L	53.96 (±1.26)	54.65 (±1.48)	46.60 (±0.91)	51.73 (±1.22)
	Parallel-13L (ours)	**58.39** (±0.51)	**55.77** (±0.82)	**46.74** (±3.27)	**53.63** (±1.53)
0%	Single-13L	**32.58** (±4.21)	12.10 (±4.66)	**26.55** (±0.65)	**23.75** (±3.17)
	Parallel-13L (ours)	27.46 (±3.60)	**18.61** (±0.27)	23.53 (±4.77)	23.20 (±2.88)

Table 17: EthioEmo benchmarking results under a limited target-language data setting. 'amh', 'orm', and 'tir' denote Amharic, Oromo, and Tigrinya, respectively.

#data	Tokenizer	hau	ibo	kin	sun	swa	avg
50%	Single-13L	52.92 (±1.09)	55.41 (±0.32)	39.57 (±2.08)	58.18 (±1.78)	18.95 (±1.33)	45.00 (±1.32)
	Parallel-13L (ours)	**57.36** (±0.62)	**56.64** (±0.95)	**41.72** (±0.83)	57.89 (±1.38)	**21.96** (±1.39)	**47.11** (±1.04)
0%	Single-3L	15.13 (±3.30)	**16.31** (±0.61)	9.72 (±0.59)	**18.93** (±2.24)	15.13 (±1.45)	15.04 (±1.64)
	Parallel-13L (ours)	**26.92** (±2.65)	16.08 (±2.15)	**23.37** (±0.32)	15.90 (±0.96)	16.38 (±3.31)	**19.73** (±1.88)

Table 18: BRIGHTER benchmarking results under a limited target-language data setting. 'hau', 'ibo', 'kin', 'sun', and 'swa' denote Hausa, Igbo, Kinyarwanda, Sundanese, and Swahili, respectively.

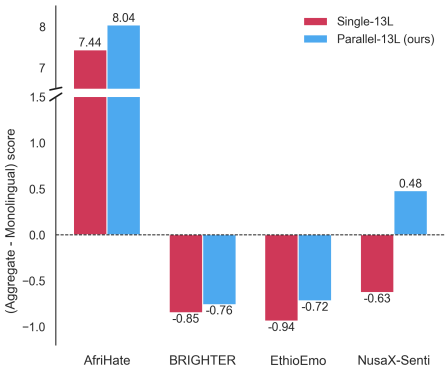

Figure 4: Performance differences across benchmarks on both cross-lingual and monolingual finetuning data for models pretrained with different tokenizer setups.

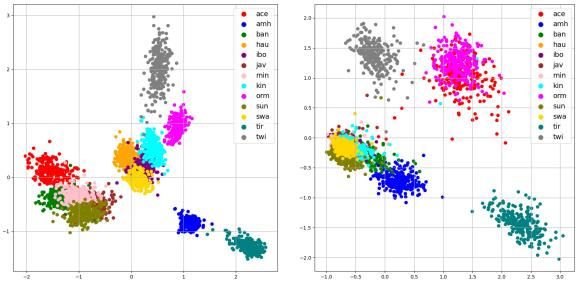

Figure 5: Principal Component Analysis (PCA) of continual pretrained model embeddings, comparing the *Single-13L* (left) and the *Parallel-13L* (right) on the parallel data FLORES+ dataset.

0.48% increase. Between *Single-13L* and *Parallel-13L*, the latter benefits more consistently from multilingual concatenation, suggesting that it leverages shared cross-lingual information more effectively. Detailed results for monolingual fine-tuning are provided in Appendix F, showing that overall performance is comparable, though *Parallel-13L* maintains slightly higher scores on average.

F Monolingual Benchmarking Details

The benchmarking results using monolingual training data are shown in Tables 19, 20, 21, and 22. Each experiment was repeated three times with preselected seeds, and we report the averages along with the standard deviations for each language. Overall, the results are comparable: *Single-13L* performs better on the NusaX-senti and AfriHate benchmarks, while *Parallel-13L* performs better on emotion classification tasks such as EthioEmo and BRIGHTER. On average across all benchmarks, however, *Parallel-13L* outperforms *Single-13L* by 0.42%.

G Continual Pre-Training: Representation Similarity

As discussed in Section 5.5, although the benchmarking performance of *Single-13L* and *Parallel-13L* after CPT is comparable, the cross-lingual representation similarity is stronger in *Parallel-13L*. This is illustrated in Figure 5, where we visualize the embedding spaces of FLORES+ (Costa-Jussà et al., 2022) sentences for each language in two dimensions using PCA. In *Single-13L*, inputs from the same language family remain clustered. For instance, Indonesian local languages (Acehnese, Minangkabau, Sundanese, etc.) cluster together, African languages cluster together, and Amharic-script languages (Amharic and Tigrinya) are separated into distinct clusters. In contrast, *Parallel-13L* shows tighter cross-lingual clustering, with languages grouped more uniformly, including Amharic. The main outliers are Tigrinya and Twi, which remain separated in both models, likely due to limited pretraining resources.

These observations are further supported by our bitext mining analysis on the models after CPT. As shown in Table 25, *Parallel-13L* consistently achieves the lowest error rates and the highest number of best `xsim` scores, outperforming the other models, although the gains are smaller than those observed when pretraining from scratch. Tables 23 and 24 provide the detailed `xsim` scores for each language pair in both *Single-13L* and *Parallel-13L*.

H Model Input Representation Design

Since we require a 'language token' to select the appropriate tokenizer for a given input, our input representation differs slightly from the conventional approach. During the model design phase, as presented in Figure 6, we explored several alternatives and ultimately selected the most effective method. Specifically, we considered three designs: (1) placing the language token at the beginning of every input sequence, requiring the model to process it as the first token; (2) the *Parallel-13L* approach, where the language token serves as a signal that is added to each token embedding, effectively combining token and language embeddings; and (3) augmenting only the embeddings of unaligned vocabulary items (i.e., tokens not belonging to the `word`-type category[8]).

[8]We store unaligned vocabulary items at the end of the vocabulary indices

Tokenizer	ace	ban	jav	min	sun	avg
Single-13L	**74.86** (±**1.02**)	**77.01** (±**0.80**)	**77.92** (±**1.62**)	77.14 (±1.00)	**76.65** (±**2.51**)	**76.72** (±**1.39**)
Parallel-13L (ours)	73.27 (±1.18)	73.76 (±0.43)	77.75 (±0.44)	**77.85** (±**0.43**)	75.76 (±2.70)	75.68 (±1.04)

Table 19: NusaX-senti benchmarking results under monolingual setting. 'ace', 'ban', 'jav', 'min', and 'sun' denote Acehnese, Balinese, Javanese, Minangkabau, and Sundanese, respectively.

Tokenizer	amh	hau	ibo	kin	orm	swh	tir	twi	avg
Single-13L	**60.92** (±**2.41**)	74.39 (±0.65)	84.96 (±0.83)	**63.82** (±**1.20**)	**73.60** (±**1.29**)	90.13 (±0.18)	61.76 (±1.66)	**56.49** (±**3.00**)	**62.17** (±**1.75**)
Parallel-13L (ours)	60.39 (±0.89)	**75.35** (±**1.52**)	**87.70** (±**0.26**)	62.94 (±1.54)	73.59 (±0.41)	**90.30** (±**0.42**)	**61.94** (±**2.77**)	55.20 (±1.15)	61.76 (±1.73)

Table 20: AfriHate benchmarking results under monolingual setting. 'amh', 'hau', 'ibo', 'kin', 'orm', 'swh', 'tir', and 'twi' denote Amharic, Hausa, Igbo, Kinyarwanda, Oromo, Swahili, Tigrinya, and Twi, respectively.

Tokenizer	amh	orm	tir	avg
Single-13L	58.70 (±1.32)	**60.15** (±**0.75**)	48.92 (±1.24)	55.92 (±1.10)
Parallel-13L (ours)	**60.98** (±**1.09**)	59.63 (±1.63)	**52.58** (±**0.20**)	**57.73** (±**0.97**)

Table 21: EthioEmo benchmarking results under monolingual setting. 'amh', 'orm', and 'tir' denote Amharic, Oromo, and Tigrinya, respectively.

Tokenizer	hau	ibo	kin	sun	swa	avg
Single-13L	57.90 (±1.40)	58.89 (±1.53)	43.89 (±0.20)	**61.05** (±**0.93**)	23.81 (±1.14)	49.11 (±1.04)
Parallel-13L (ours)	**60.23** (±**0.35**)	**59.23** (±**0.71**)	**47.74** (±**2.45**)	60.97 (±0.79)	**24.04** (±**2.15**)	**50.44** (±**1.29**)

Table 22: BRIGHTER benchmarking results under monolingual setting. 'hau', 'ibo', 'kin', 'sun', and 'swa', denote Hausa, Igbo, Kinyarwanda, Sundanese, and Swahili, respectively.

	ace	amh	ban	hau	ibo	jav	min	kin	orm	sun	swa	tir	twi
ace		**91.01**	**65.42**	68.28	**76.09**	**55.83**	**67.59**	**82.61**	**93.68**	79.05	68.28	**97.73**	**82.51**
amh			93.48	90.42	89.72	88.24	90.51	91.90	**94.96**	89.92	90.81	92.29	94.76
ban				**40.02**	**50.40**	**27.96**	**44.86**	74.51	91.40	50.59	38.54	97.33	77.08
hau					36.26	29.25	41.80	67.00	90.61	**40.32**	24.90	95.55	69.76
ibo						**43.68**	**55.14**	74.11	92.59	**53.56**	40.51	**96.44**	76.38
jav							24.41	70.85	91.70	**28.06**	29.05	95.65	75.30
min								**69.96**	90.51	**28.36**	38.14	96.54	75.30
kin									94.37	**73.62**	66.21	**96.84**	85.57
orm										**91.11**	90.81	**97.43**	**93.08**
sun											**33.10**	96.05	**75.79**
swa												95.16	73.22
tir													98.02
twi													

Table 23: Bitext mining metric scores on CPT models trained with *Single-13L* tokenizer.

	ace	amh	ban	hau	ibo	jav	min	kin	orm	sun	swa	tir	twi
ace		98.02	77.37	**62.55**	80.53	57.41	90.32	98.62	97.83	**68.58**	**59.98**	98.52	89.43
amh			**74.90**	**62.65**	**78.46**	**63.54**	**64.62**	**76.88**	95.06	**71.15**	**59.49**	97.33	**86.07**
ban				41.01	61.07	30.43	56.82	**63.34**	**85.18**	**37.06**	**34.19**	**92.98**	**76.68**
hau					**31.82**	**20.06**	**28.75**	**47.92**	**79.15**	53.56	**17.39**	**90.61**	**63.24**
ibo						43.97	69.86	**66.60**	**85.18**	62.35	**36.76**	97.04	**72.63**
jav							**22.43**	**50.79**	**76.98**	52.87	**22.13**	**92.69**	**69.57**
min								76.98	**89.53**	57.21	**25.10**	**92.79**	**72.23**
kin									**91.70**	83.60	**54.64**	97.04	**78.56**
orm										96.34	**80.53**	98.81	98.22
sun											40.32	**90.12**	85.28
swa												**91.21**	**68.08**
tir													**97.92**
twi													

Table 24: Bitext mining metric scores on CPT models trained with *Parallel-13L* tokenizer.

	Single-13L	Parallel-13L
Avg. `xsim`↓	72.18	**69.34**
# of best `xsim`↑	29	**48**

Table 25: Bitext mining metric meta-scores on CPT models.

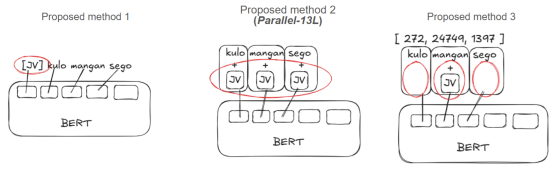

Figure 6: The design of the model input representation. We adopt the second method and refer to it as *Parallel-13L*.

We pretrained models with each of these methods from scratch using the same dataset (Wikipedia dumps for the 13 languages studied in this paper). The first and third approaches did not converge as quickly or effectively as *Parallel-13L*, which therefore became our chosen design for the experiments reported in Section 5 and beyond.