MoVoC: Morphology-Aware Subword Construction for Ge’ez Script Languages

Anonymous ACL submission

Abstract

Subword-based tokenization methods often fail to preserve morphological boundaries, a limitation especially pronounced in low-resource, morphologically complex languages such as those written in the Ge‘ez script. To address this, we present **Mo**rpheme-aware Subword **Vo**cabulary **C**onstruction MoVoC and train MoVoC-Tok, a tokenizer that integrates supervised morphological analysis into the subword vocabulary. This hybrid segmentation approach combines morpheme-based and Byte Pair Encoding (BPE) tokens to preserve morphological integrity while maintaining lexical meaning. To tackle resource scarcity, we curate and release manually annotated morpheme data for four Ge‘ez script languages and a morpheme-aware vocabulary for two of them. While the proposed tokenization method does not lead to significant gains in automatic translation quality, we observe consistent improvements in intrinsic metrics, MorphoScore, and Boundary Precision, highlighting the value of morphology-aware segmentation in enhancing linguistic fidelity and token efficiency. Our morpheme-annotated datasets and tokenizer dataset will be publicly available to support further research in low-resource, morphologically rich languages.

1 Introduction

Tokenization is a fundamental preprocessing step in NLP, converting raw text into structured units such as bytes (Gillick et al., 2016), characters (Al-Rfou et al., 2019), subwords (Sennrich et al., 2016), words (Song et al., 2021), or multi-word expressions (Gee et al., 2023). Subword tokenization, such as BPE (Sennrich et al., 2016), gained popularity for being language-independent and compressing vocabulary, enabling efficient and balanced token learning. However, subwords often fail to capture morphological structure, a problem that is especially clear in multilingual models using a shared vocabulary across languages. Without careful and balanced data selection, low-resource languages typically receive fewer subwords, leading to a high token-to-word ratio (Haddow et al., 2022; Limisiewicz et al., 2023; Libovický and Helcl, 2024). Moreover, negative morphemes challenge LLMs with tokenizers lacking morphological sensitivity (Mikaberidze et al., 2024).

Morphological systems across languages vary regarding the relation between form and meaning (Socolof et al., 2022) and are harder to model and predict (Cotterell et al., 2018; Park et al., 2021). Furthermore, languages with greater morphological complexity may lead to worse language model performance, as morphologically rich languages tend to have a large number of very infrequent word forms produced by combinations of morphemes, which leads to data sparsity (Shin and You, 2009; Botev et al., 2022). Morphologically rich languages also have less annotated data (Botev et al., 2022) and are often written with non-Latin scripts, which require more bytes to be represented in common encoding standards like UTF-8 (Arnett and Bergen, 2025).

Additionally while subword tokenization consistently outperforms character and word level approaches by efficiently compressing text into shorter token sequences (Goldman et al., 2024), compression alone is not always predictive of downstream task success, especially for morphologically complex or semantically dense languages (Schmidt et al., 2024). Overall the standard algorithms lack morphology awareness (Libovický and Helcl, 2024). Alternatively *Morphologically-aware* tokenization produces more meaningful tokens, often resulting in improved model performance (Lerner and Yvon, 2025; Hofmann et al., 2022; Bauwens and Delobelle, 2024; Minixhofer et al., 2023). This claim is supported by evidence from several languages, e.g., Ko-

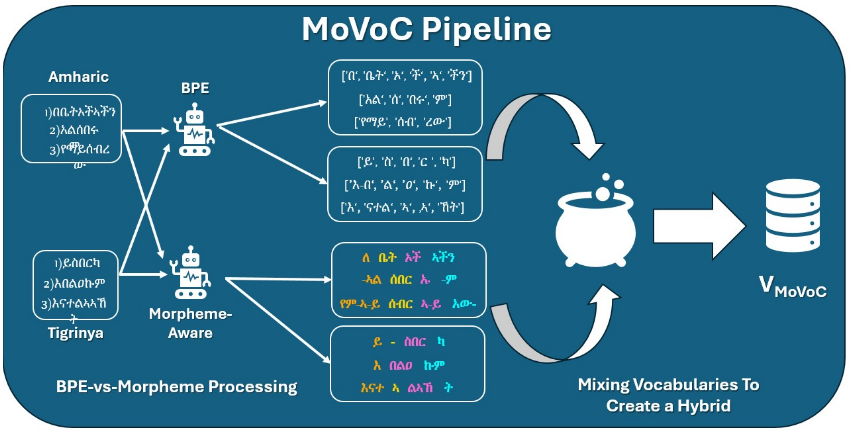

Figure 1: MoVoC Pipeline. We first extract Amharic and Tigrinya words from our corresponding text corpora to perform token-based and morpheme-based separation resulting in four different vocabularies. We then merge all four vocabularies to generate a single MoVoC-based vocabulary (V_{MoVoC}).

rean (Lee et al., 2024), Arabic (Tawfik et al., 2019), Japanese (Bostrom and Durrett, 2020), and Hebrew (Gueta et al., 2023). However, such efforts are constrained by the limited availability of morphologically annotated datasets, mostly found in high-resource languages (Minixhofer et al., 2023). Moreover, even with sufficient morphological annotations, relying solely on morpheme-based tokenization is suboptimal. As noted by Bauwens and Delobelle (2024), approaches that disregard subword tokenization struggle to leverage shared statistical information across related word forms, result in longer input sequences, and generally perform worse than hybrid methods that integrate morphological information with subword tokenization. Accordingly, based on the three-stage tokenization framework proposed by Schmidt et al. (2024) and Libovický and Helcl (2024), we adopt a tokenization process comprising three distinct phases: pre-tokenization, vocabulary construction, and segmentation.

Concretely, our contributions are as follows: (i) We develop morphologically annotated datasets for four low-resource Ge'ez Script languages to support improved tokenization, morphological analysis, and downstream NLP tasks; (ii) We propose MoVoC (**Mo**rpheme-Aware **Vo**cabulary **C**onstruction), a supervised approach that leverages linguistically informed BPE segmentations as an alternative to standard subword techniques. It integrates BPE-derived vocabulary with morphemes extracted through supervised methods to enhance morphological representation and improve token-to-morpheme alignment in morphologically rich low-resource languages; (iii) We perform thorough intrinsic and extrinsic evaluations to measure the effectiveness of our approach and validate its practical applicability.

2 Background

2.1 Tokenization Approaches

Word Tokenization: The straightforward methods for segmenting text involves breaking a string of text into distinct words. The simplest form of *word* tokenization relies on dividing sequences by whitespace and treating each word as a separate token, as noted by (Bengio et al., 2003). This method is particularly prevalent in languages that have clear word boundaries, such as English. Alternatively, one could also perform *rule-based* tokenization, which uses a set of predefined rules and patterns to identify tokens. For example, it might use regular expressions to handle contractions like "can't" or "won't" by splitting them into "can not" and "will not," respectively. A key challenge with this method is addressing out-of-vocabulary (OOV) words, which can arise from typos, unrecognized scripts, and other factors.

Subword Tokenization: The most common strategy is to decompose words into subwords, allowing models to process out-of-vocabulary words by merging subwords from the vocabulary (Kudo and Richardson, 2018). Examples of popular subword tokenizers are WordPiece (Song et al., 2021), BPE (Sennrich et al., 2016), Byte-Level BPE (BBPE) (Wang et al., 2020) and Unigram (Kudo and Richardson, 2018).

Character Tokenization: Tokenization can also be done at the character or UTF-8 byte level, but this increases sequence length and leads to higher computational cost due to the quadratic complexity of transformers’ self-attention (Vaswani et al., 2017; Ali et al., 2024).

2.2 Ge’ez Script Natural Language Processing

The Ge’ez writing system is one of the oldest continuously used scripts in the world, which preserved for over 2000 years, it reflects a remarkably stable and adaptable method of representing language (Gidey et al., 2024). Beyond its function as a grammatical system, Ge’ez offers valuable insight into the intellectual, philosophical, and cultural foundations of ancient African civilizations, highlighting their linguistic innovation and societal advancement (Scelta and Quezzaire-Belle, 2001; Gidey et al., 2024). Unlike alphabetic systems like Greek or Latin, Ge’ez uses a syllabic script, originally derived from the Sabean script (Bekerie, 2003). However, the development and publication of usable NLP tools for Ge’ez have been hindered to date due to the scarcity of essential linguistic resources (Gidey et al., 2024). It is a script for numerous languages, including Ge’ez, Tigrinya, Amharic, Tigre, and Blin (Gaim et al., 2022). These morphologically rich Semitic languages present unique difficulties due to their complex morphology, which generates numerous inflected forms (Tedla and Yamamoto, 2018).

For example, the Amharic verb ”to write” (ጻፈ) inflects for tense, aspect, person, number, and gender: ”I wrote” becomes ጻፍኩ, ”you wrote” ጻፍክ, and ”they wrote” ጻፉ. Similarly, in Tigrinya, the root verb ”to write” (ጸሓፈ) exhibits rich inflectional patterns based on similar grammatical categories. This rich inflectional morphology results in many surface forms, especially because verb endings are often agglutinated directly to the root without clear word boundaries. For example, in Amharic, the root verb ሠራ (”to do”) forms inflected variants by attaching endings directly, such as ሠራሁ (”I did”). Similarly, in Tigrinya, the root verb ነበረ (”to be” or ”to exist”) forms inflected variants like ነበረኒ (”I had it”) by attaching endings directly to the root. Consequently, subword tokenization methods such as BPE, which are based solely on frequency and not linguistic structure, struggle to segment these languages appropriately. In this work, we contribute to Ge’ez script language processing by (i) providing annotated data for four Ge’ez script language, such as Tigriyna, Amharic, Ge’ez and Tigre, and (ii) exploring a morpheme-aware alternative to BPE by constraining segmentation to respect morpheme boundaries, yielding vocabularies aligned with linguistic structure and improving tokenization in morphologically rich languages.

2.3 Tokenization for Morphologically Rich and Low-Resource Languages

Subword tokenization is a widely explored area in natural language processing, with various methods proposed to break words into smaller subword units (Hou et al., 2023; Gezmu and Nürnberger, 2023; Socolof et al., 2022; Dewangan et al., 2025; Thawani et al., 2023; Schmidt et al., 2024). Park et al. (2021) trained models with several segmentation algorithms, including BPE and Morfessor (Creutz and Lagus, 2002), on a corpus of Bible verses in 92 languages. Language models perform worse on morphologically complex languages due to poor tokenization and smaller datasets (Arnett and Bergen, 2025). Morphological typology significantly impacts LM performance, with features like exponence, flexivity, and fusion contributing to low-frequency phenomena that are challenging for statistical models (Gerz et al., 2018).

Park et al. (2021) investigate linguistically motivated segmentation methods, such as Morfessor and FSTs, that help reduce the effect of morphological complexity and can improve language modeling performance. Dewangan et al. (2025) show that optimized BPE outperforms greedy methods by reducing token count, improving efficiency, and benefiting multilingual and low-resource NLP tasks. Similarly, both MorphBPE (Asgari et al., 2025) and MorphPiece (Jabbar, 2023) aim to improve tokenization for morphologically rich languages by incorporating linguistic features into the segmentation process. Mikaberidze et al. (2024) explored the impact of various tokenization methods on Georgian language modeling, demonstrating the necessity of preserving morphological variations in the tokenization process. Furthermore, Bauwens and Delobelle (2024) concluded that a lack of morpheme-awareness leads to inconsistent intraword representations, inflated vocabulary size, and inefficient embedding storage. On the other hand, Saleva and Lignos (2021) evaluate sub-

Original Words Amharic / English	**Morphological Segmentation**	**BPE Segmentation**	**Impacts of BPE Segmentation**
በቤትኦችኣችን In our homes	**'ለ-<ቤት>ኦች-ኣችን--'**	['**በ**','**ቤት**','**ኦ**','**ች**','**ኣ**','**ችን**']	Over-segmentation: BPE splits the phrase into fragments that don't form the cohesive meaning of ”In our homes.” Words like '**ቤት**' (house) and '**ኦችኣችን**' (our homes) are broken, losing their semantic coherence and making it hard to convey the original context.
አልሰበሩም They didn't break	**’-አል-<ሰበር>ኡ--ም---’**	['**አል**','**ሰ**','**በሩ**','**ም**']	Fragmented meaning: BPE breaks ”**ሰበር**” (break) into ”**ሰ**” and ”**በሩ**” distorting the intended meaning. The sentence ”They didn't break” becomes unclear because BPE doesn't preserve the root structure of the verb, leading to loss of grammatical clarity and reduced accuracy in translation.
የማይሰብረው He didn't break	**’የም-ኣ-ይ<ሰብር>-አው----’**	['**የማይ**', '**ሰብ**', '**ረው**']	Loss of verb tense and context: BPE splits ”**ሰብር**” (break) incorrectly, making it ”**ሰብ**” and ”**ረው**.” The full context of ”He didn't break” is lost because the tense (”didn't”) and action (”break”) are not preserved in the segmentation, leading to confusion and a loss of meaning.

Table 1: Comparison of Morphological Segmentation and BPE Segmentation with their Impact. The stem is enclosed within < >, while - marks the boundaries of morphemes. A lone - indicates the absence of a morpheme in that position.

word segmentation in low-resource NMT and find that while morphology-based methods occasionally outperform BPE, their overall performance is often statistically similar, offering no consistent advantage. A more recent work by Libovický and Helcl (2024) employed word embeddings to enable semantically informed subword segmentation; however, its effectiveness diminishes in low-resource languages due to limited training data.

Overall, these reports reveal a gap in understanding the performance of different tokenization strategies for low-resource languages. Therefore, our work on MoVoC not only contributes to the processing of Ge'ez script languages but also acts as another successful example for the potential of morpheme-aware subword tokenization alternative.

3 Proposed Method

In this section, we describe three separate methodologies for 1) Pre-tokenization and Supervised Morphological Analyses, 2) Vocabulary Construction (MoVoC), and 3) Morpheme-aware Subword Segmentation (MoVoC-Tok).

3.1 Pre-tokenization and Supervised Morphological Analyses

Pre-tokenization is a preparatory step before tokenization, segmenting raw text into manageable units. Unlike high-resource languages like English, Ge'ez-script languages lack robust NLP tools, making basic preprocessing tasks such as stopword removal, punctuation normalization, and special character filtering challenging. To address this gap, we develop a pre-tokenization pipeline based on customized regular expressions tailored to the orthographic and morphological characteristics of Ge'ez-script languages. The pipeline begins with corpus cleaning to eliminate noise and inconsistencies, followed by supervised morphological analysis for accurate morpheme extraction. For Amharic and Tigrinya, we leverage HornMorpho[1], a rule-based morphological analyzer. While HornMorpho performs reliably on Amharic, its Tigrinya outputs often require manual post-editing due to limited coverage. For languages without existing analyzers, Ge'ez and Tigre, we manually construct and annotate morphemes under linguistic supervision. This morpheme-level annotation is critical for high-quality segmentation, particularly in morphologically rich constructions. As shown in Tab. 1, naïve application of Byte Pair Encoding (BPE) frequently results in over-segmentation or incorrect splits that obscure grammatical and semantic content. For instance, the Amharic word **አልሰበሩም** (”They didn't break”) is fragmented by BPE into **አል**, **ሰ**, **በሩ**, and **ም**, breaking the verb stem **ሰበር** and distorting meaning **በሩ** alone is a valid word meaning ”gate.” These failures are attributed to BPE's unsupervised, morphology-agnostic nature, which ignores linguistic boundaries and often produces incoherent subword units. Such misalignments are especially detrimental for low-resource,

[1] https://github.com/hltdi/HornMorpho

morphologically complex languages where training data is scarce. To mitigate this, we introduce a supervised morphological segmentation approach integrated into a linguistically informed pre-tokenization stage. This ensures morphemes are extracted and preserved before applying subword algorithms like MoVoC in Algorithm 1, aligning tokenization with true morphological structures and enhancing both semantic coherence and downstream NLP utility.

The annotated morphemes serve as a gold-standard *test set* for evaluating token-to-morpheme boundary alignment and also intended to support future research as a publicly available benchmark for morphological segmentation evaluation.

3.2 Vocabulary Construction (MoVoC)

Existing subword tokenization methods primarily rely on statistical analysis of occurrence frequencies without explicitly considering morphemes (Truong et al., 2024). The most popular method, Byte-Pair Encoding (BPE) (Sennrich et al., 2016), greedily merges the most frequent token pairs to form subword units. Similarly, the Unigram Language Model, as implemented in SentencePiece (Kudo and Richardson, 2018), selects high-probability segmentations using a probabilistic unigram model. However, these methods often suffer from limited morphological generalization, which can negatively impact interpretability, compositionality, and cross-lingual transfer especially in morphologically rich languages.

In this work, we target Ge‘ez script languages, which are characterized by fusional morphology and a scarcity of linguistically motivated tools and morphologically annotated data.

As highlighted in the pre-tokenization analysis (see Sec. 3.1), we employ a supervised morphological analysis. Based on the annotated morphemes, we design a hybrid vocabulary construction strategy as we described see Algorithm 1 (see Tab. 5 for the resulting vocabulary size). Our algorithm allocates a predefined portion of the total vocabulary to morpheme units and the rest to BPE tokens, balancing linguistic structure and statistical efficiency. This integration of morpheme-aware subwords and BPE-based tokens ensures better vocabulary control, reduces the incidence of rare tokens, and preserves semantic granularity in subword representations.

Methodology: Let P_{am} represent the Amharic monolingual corpus, and P_{ti} represent the Tigrinya monolingual corpus. The goal of the MoVoC method is to create a final vocabulary V_{MoVoC} that combines subword tokenization from the BPE model and morpheme-based tokenization, with an emphasis on incorporating a higher proportion of morpheme-based tokens in the vocabulary rather than employing the BPE model derived from the two corpora. The vocabulary V_{MoVoC} is formulated by merging the vocabularies obtained from the BPE model and the morpheme token set. When the target vocabulary size for $\text{BPE}_{\text{MoVoC}}$ is s, we train $\text{BPE}_{\text{small}}$ with a vocabulary size of $s(1-r)$ where r is a hyperparameter set between 0 and 1 to denote the proportion of added morpheme tokens in V_{MoVoC}.

Algorithm 1 MoVoC Pseudocode

```
Require:
  $P_{am}$ (Amharic corpus),
  $P_{ti}$ (Tigrinya corpus),
  $s$ (Total vocabulary size),
  $r$ (Proportion of morpheme-aware tokens, $0 \leq r \leq 1$)
Ensure: $V_{\text{MoVoC}}$ (Final MoVoC vocabulary)
  Step 1: Perform Morpheme Segmentation using HornMorpho
  $M_{am} \leftarrow$ HornMorpho_segment($P_{am}$)
  $M_{ti} \leftarrow$ HornMorpho_segment($P_{ti}$)
  Step 2: Define Vocabulary Sizes
  $s_{\text{lang}} \leftarrow s/2$
  $s_{\text{BPE}} \leftarrow s_{\text{lang}} \times (1-r)$
  $s_{\text{morpheme}} \leftarrow s_{\text{lang}} \times r$
  Step 3: Train BPE Models
  $V_{\text{BPE},am} \leftarrow$ Train_BPE($P_{am}, s_{\text{BPE}}$)
  $V_{\text{BPE},ti} \leftarrow$ Train_BPE($P_{ti}, s_{\text{BPE}}$)
  Step 4: Extract Morphemes
  $V_{\text{morpheme},am} \leftarrow$ extract_morphemes($P_{am}, s_{\text{morpheme}}$)
  $V_{\text{morpheme},ti} \leftarrow$ extract_morphemes($P_{ti}, s_{\text{morpheme}}$)
  Step 5: Merge All Vocabularies
  $V_{\text{MoVoC}} \leftarrow V_{\text{BPE},am} \cup V_{\text{BPE},ti} \cup V_{\text{morpheme},am} \cup V_{\text{morpheme},ti}$
  Step 6: Train Final MoVoC Model
  Train_MoVoC_Model($V_{\text{MoVoC}}$)
  Step 7: Return Final Vocabulary
  return $V_{\text{MoVoC}}$
```

MoVoC Hyperparameter Setting: Hyperparameter tuning plays a crucial role in vocabulary construction using BPE and other subword tokenization techniques, especially in morpheme-aware settings. In morphologically rich fusional languages, words often consist of multiple morpheme roots, prefixes, and suffixes. Without careful tuning, BPE may (i) overfit to whole words, missing productive morphemes (e.g., **ኦች**, **-ኦችን**), or (ii) ignore language-specific morphological structures, as seen in verb forms like **እሄዳለሁ** and

ትሄዳለህ, where affixes encode subject agreement. Proper tuning ensures subword units align with meaningful morphemes, improving linguistic representation and downstream model performance.

3.3 MoVoC-Tok (Morpheme-aware Subword Segmentation)

We train the BPE tokenizer using the mixed vocabulary obtained from MoVoC and subsequently assess its performance. However, despite employing a MoVoC-derived vocabulary, a conventional BPE tokenizer may still produce morpheme boundary violations, as its merge operations are data-driven and can combine subwords that cross morpheme boundaries if not explicitly restricted. To address this, we incorporate morphological constraints directly into the BPE training process by limiting merge candidates to those that do not span morpheme boundaries defined by MoVoC. This integration of morphological information ensures that the resulting tokenization (MoVoC-Tok) adheres to true morphological segmentation, thereby preventing invalid merges.

Let $W = \{w_1, w_2, \ldots, w_n\}$ be the vocabulary of words obtained from MoVoC where each word w_i is a sequence of characters $w_i = (c_1, c_2, \ldots, c_m)$. Let $M_i = \{b_1, b_2, \ldots, b_k\}$ be the morpheme boundaries in w_i, as provided by MoVoC in Sec. 3.2. Then, Morpheme-Aware BPE Segmentation can be formally defined as follows:

$$\max_{V} \sum_{w_i \in W} \log P(\mathrm{BPE}(w_i; V, M_i)),$$

where the following constraint holds:

$$\mathrm{BPE}(w_i; V, M_i) = (s_1, s_2, \ldots, s_t)$$

such that

$$\forall s_j,\ s_j \subseteq w_i \text{ and } s_j \text{ does not cross } M_i.$$

Here, V denotes the learned subword vocabulary and s_j represents BPE merge units that are constrained by the morpheme boundaries M_i. The merge operations are further restricted such that

$$(a, b) \in \mathrm{MergeCandidates} \Rightarrow a \cup b \notin M_i^{\complement}.$$

In other words, merges are permitted only if they **do not cross** morpheme boundaries as defined by MoVoC in Sec. 3.2.

4 Experimental Setup

4.1 Target Languages

We focus on four languages that use the Geez script: Amharic, Tigrinya, Ge'ez, and Tigre. These languages exhibit rich and complex morphological structures, posing significant challenges for conventional subword segmentation methods like BPE.

Amharic and Tigrinya: We perform morpheme segmentation using the HornMorpho analyzer, which decomposes words into stems and affixes. These segmented units are used both for vocabulary construction.

Ge'ez and Tigre: Due to the absence of analyzers and corpora, we perform manual morpheme annotation using expert linguistic guidelines. These annotations are applied for testing purposes only and are not part of the vocabulary since we did not get data for BPE training.

4.2 Dataset Details

Training Data: We have trained BPE to create the subword vocabularies in addition to Morphemes, and we finetuned Machine Translation as a downstream task. For these, we use the HornMT[2] corpus as the primary source for annotating the morpheme and the NLLB project (Costa-Jussà et al., 2022) for training BPE to construct the vocabularies in Tigrinya and Amharic. For the finetuning model we use parallel corpora mined and released by Meta AI as part of the No Language Left Behind (NLLB) project (Costa-Jussà et al., 2022). Specifically, we employ the English–Tigrinya and English–Amharic and vise versa parallel corpora to assess machine translation performance. These datasets were created using the stopes mining library and LASER3 encoders (Costa-Jussà et al., 2022), providing high-quality mined bitext for 148 English-centric and 1465 non-English-centric language pairs. Due to the noisy nature of the mining process, we utilize this data solely for model training.

Evaluation Data: Amharic and Tigrinya: Both languages are directly supported by Flores-200 (Goyal et al., 2022). We use the corresponding development and test sets for automatic evaluation using BLEU (Papineni et al., 2002) and chrF++ (Popović, 2017). But since Ge'ez and Tigre were not included in the FLORES-200 (Goyal

[2] https://github.com/asmelashteka/HornMT

Language (ISO 639-3)	No. Items	MorphScore ↑
Amharic (amh)	80k	0.71
Tigrinya (tir)	80k	0.731
Ge’ez (gez)	20k	0.67
Tigre (tig)	32k	0.654

Table 2: Languages for which we created morphological datasets with the corresponding **MoVoC-Tok** tokenizer’s MorphScore (which we want to maximize, indicated by ↑). All four languages are Afro-Asiatic and Semitic, written in Ge’ez script, and utilize fusional morphemes.

et al., 2022) benchmark and were not part of the finetuning data (Costa-Jussà et al., 2022), we finally consider 100 sentence pairs from the OPUS parallel corpus (Tiedemann, 2012) as a final evaluation for all languages.

Test Data: Extrinsic evaluation was conducted using an unseen subset of the first 100 sentence pairs from the OPUS parallel corpus (Tiedemann, 2012) for each target language: Amharic, Tigrinya, and Tigre. Due to the absence of parallel data, Ge‘ez was evaluated only intrinsically. For all languages, intrinsic evaluation was based on our annotated morphemes test set, designed to analyze segmentation quality.

4.3 Training Setup and Configuration

We trained our tokenizer using the Hugging Face tokenizers library (Wolf et al., 2020) and analyze BPE, WordPiece, as baseline subword tokenizers, using the mplementations from HuggingFace[3]. And we fine-tuned the MarianMT (Junczys-Dowmunt et al., 2018) transformer model on a single NVIDIA GPU using a Slurm-managed HPC cluster. The job requested 1 GPU, 6 CPU cores, 32 GB of RAM, and a maximum runtime of 24 hours. The training environment was managed via Conda for reproducibility. Training was performed for 3 epochs with a batch size of 8 and a maximum sequence length of 128 tokens and transformers version: ”4.51.3”. The learning rate started at 1.44×10^{-7} and decayed throughout training. Gradient norms decreased from 1.14 to 1.06, and the training loss ranged from 0.443 to 0.438 across epochs.Training time was approximately 12 hours, with an average speed of 96.7 samples per second.

5 Evaluation Framework

We incorporate both intrinsic and extrinsic evaluations to assess our approach. Intrinsic evaluation focuses on morpheme boundary precision and vocabulary consistency (e.g., Rényi entropy), while extrinsic evaluation measures downstream performance in machine translation using metrics like BLEU and chrF++.

5.1 Extrinsic Evaluation

Translation quality is assessed using BLEU and chrF++, which measure n-gram and character-level overlap. However, as these metrics may overlook morphological improvements, we complement them with intrinsic evaluations for a more complete analysis.

Machine Translation As part of the downstream evaluation of our framework, we present a fine-tuned MarianMT (Multilingual Transformer) model for machine translation between English and two low-resource Ge’ez script languages: Amharic and Tigrinya. The model was trained on parallel corpora consisting of English-Amharic and English-Tigrinya sentence pairs. Although Tigre was not included during training, it was incorporated in the evaluation phase to assess the model’s zero-shot translation capabilities. The model architecture consists of 6 encoder and 6 decoder layers, each with 8 attention heads and a hidden size of 512. It employs a feedforward dimension of 2048, Swish activation, shared encoder-decoder embeddings, and static positional encodings. The vocabulary size is 63,050 tokens. All training and evaluation were conducted using the Hugging Face Transformers library (version 4.51.3). This work serves as a benchmark for future research in low-resource neural machine translation involving Ge’ez script languages.

5.2 Intrinsic Evaluation

To get a better understanding of how well different tokenization strategies preserve morphemes, we measure the alignment between BPE tokens and gold-standard morphemes using Morpheme boundary precision (Nouri and Yangarber, 2016) and MorphScore (Arnett and Bergen, 2025).

Morpheme boundary precision: This form of precision is a traditional metric from morphological segmentation, where all predicted boundaries (across all words) are compared to gold-standard boundaries.

[3] https://github.com/huggingface/tokenizers

Strategy	BLEU ↑	chrF++ ↑
English → Amharic		
BPE	0.2150 ± 0.0120	16.2000 ± 1.05
WordPiece	0.2340 ± 0.0155	16.5000 ± 1.00
MoVoC-Tok	**0.2455 ± 0.0108**	**17.8500 ± 0.95**
English → Tigrinya		
BPE	0.1720 ± 0.0095	7.2000 ± 0.85
WordPiece	0.1880 ± 0.0088	7.5000 ± 0.80
MoVoC-Tok	**0.2050 ± 0.0080**	**8.1000 ± 0.75**
English → Tigre		
BPE	0.0950 ± 0.0080	4.0000 ± 0.70
WordPiece	0.1025 ± 0.0075	4.3000 ± 0.65
MoVoC-Tok	**0.1175 ± 0.0068**	**5.1500 ± 0.60**
English → Ge’ez		
BPE	0.0480 ± 0.0070	3.0500 ± 0.55
WordPiece	0.0550 ± 0.0065	3.2500 ± 0.60
MoVoC-Tok	**0.0660 ± 0.0060**	**3.9500 ± 0.50**

Table 3: Translation performance of BPE, WordPiece, and MoVoC-Tok for English to Amharic, Tigrinya, Tigre, and Ge’ez. Metrics are reported as mean ± standard deviation over multiple runs. Best scores per language are bolded.

MorphScore: We compute MorphScore by assigning 1 if a token boundary aligns with the gold morpheme boundary, and 0 otherwise. Unsegmented words (i.e., full matches in the vocabulary) are excluded. As you can see in Tab. 2, the final MorphScore is the mean of these values across our morpheme test set.

Rényi entropy: The Rényi entropy (Rényi, 1961) over token distributions quantifies subword diversity and balance, where lower values indicate sharper and more consistent segmentation, reflecting clearer morpheme boundaries, while higher values suggest ambiguity or uncertainty in token boundary placement.

6 Result

Tokenization Quality. MoVoC-Tok achieves MorphScores for all four languages (see Tab. 2) that are substantially higher than the mean MorphScore reported for fusional languages in the original paper of Arnett and Bergen (2025) (0.533). While MoVoC-Tok does not score higher than all SentencePiece tokenizer variants, this indicates that our hybrid approach instills at least partial morpheme awareness into the tokenization process. Our intrinsic evaluation results (see Tab. 4) further underscore this general result: generating tokens via MoVoC-Tok yields both better Rényi Entropy and Morpheme Boundary precision scores across all four languages. While the effect for Amharic and Tigrinya text is less pronounced, we can observe a clear performance boost when processing the less-represented low-resource languages, Tigre and Ge’ez.

Downstream Task Performance. To evaluate the utility of our morpheme-aware tokenizer, we investigated the machine translation (MT) performance from English to our target languages, Amharic, Tigrinya, Tigre, and Ge’ez. Table 3 presents the results for the first 100 sentences of the OPUS test set using the tokenizers BPE, WordPiece, and MoVoC-Tok. Overall, we can observe that MoVoC-Tok consistently outperforms the other tokenizers across all three translation tasks.

Language	Tokenization	Precision ↑	Rényi ↓ Entropy
Amharic	MoVoC-Tok	**85.5**	**0.40**
	BPE	85.3	0.41
Tigrinya	MoVoC-Tok	**88.3**	**0.39**
	BPE	83.9	0.40
Tigre	MoVoC-Tok	**83.9**	**0.44**
	BPE	74.6	0.49
Ge‘ez	MoVoC-Tok	**85.6**	**0.40**
	BPE	73.9	0.44

Table 4: Morpheme Boundary Precision and Rényi Entropy ($\alpha = 2$) for 32k Vocabularies across tokenization strategies. MoVoC-Tok shows improved precision and reduced entropy, indicating more accurate and consistent subword segmentation. ↑ / ↓ indicates that the metric should be maximized/minimized.

7 Conclusion and Future Work

In this work, we extend the processing of Ge’ez script languages by (i) releasing morphologically annotated datasets for four languages, Tigrinya, Amharic, Ge’ez, and Tigre, and (ii) proposing a morpheme-aware tokenization approach as an alternative to conventional BPE. Our method constrains subword segmentation to align with morpheme boundaries, resulting in vocabularies that better reflect the underlying linguistic structure and improve tokenization quality for morphologically rich languages. The annotated data will further serve for research and evolution in low-resource language processing, supporting improved linguistic analysis and more effective natural language models.

8 Limitations and Ethical Considerations

8.1 Limitations

The proposed morphology-aware tokenization approach, while improving intrinsic metrics such as MorphoScore and Boundary Precision, does not yield significant gains in automatic translation quality. The curated morpheme-annotated datasets and vocabulary are limited to a small set of Geʻez script languages, which may affect the generalizability of the method. Furthermore, the increased complexity of the hybrid tokenization approach may not translate to proportional performance improvements in downstream NLP tasks.

8.2 Ethical Considerations and Use of Resources

In this study, we utilized publicly available datasets such as NLLB, OPUS, and HornMT for training and evaluation purposes. For morphological segmentation and analysis, we employed the HornMorpho tool, a rule-based morphological analyzer designed for Horn of Africa languages. All external resources were used in alignment with their respective licenses and intended research use.

Additionally, we created and will release manually morpheme-annotated datasets and morpheme-aware vocabularies for four Ge'ez script languages: Amharic, Tigrinya, Tigre, and Ge'ez. These artifacts are intended solely for research purposes and will be made publicly available under open data licenses to support further work on low-resource, morphologically rich languages. We ensure that our use and release of all resources comply with ethical standards and usage constraints associated with their original access conditions. To enhance the readability of the manuscript, we used ChatGPT for paraphrasing and language editing.

References

Rami Al-Rfou, Dokook Choe, Noah Constant, Mandy Guo, and Llion Jones. 2019. Character-level language modeling with deeper self-attention. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):3159–3166.

Mehdi Ali, Michael Fromm, Klaudia Thellmann, Richard Rutmann, Max Lübbering, Johannes Leveling, Katrin Klug, Jan Ebert, Niclas Doll, Jasper Buschhoff, Charvi Jain, Alexander Weber, Lena Jurkschat, Hammam Abdelwahab, Chelsea John, Pedro Ortiz Suarez, Malte Ostendorff, Samuel Weinbach, Rafet Sifa, Stefan Kesselheim, and Nicolas Flores-Herr. 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.

Catherine Arnett and Benjamin Bergen. 2025. Why do language models perform worse for morphologically complex languages? In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 6607–6623, Abu Dhabi, UAE. Association for Computational Linguistics.

Ehsaneddin Asgari, Yassine El Kheir, and Mohammad Ali Sadraei Javaheri. 2025. Morphbpe: A morpho-aware tokenizer bridging linguistic complexity for efficient llm training across morphologies. *arXiv preprint arXiv:2502.00894*.

Thomas Bauwens and Pieter Delobelle. 2024. BPE-knockout: Pruning pre-existing BPE tokenisers with backwards-compatible morphological semi-supervision. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5810–5832, Mexico City, Mexico. Association for Computational Linguistics.

Ayele Bekerie. 2003. Historical overview of ethiopic writing system's possible influence on the development of the armenian alphabet. *International Journal of Ethiopian Studies*, pages 33–58.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155.

Kaj Bostrom and Greg Durrett. 2020. Byte pair encoding is suboptimal for language model pretraining. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4617–4624, Online. Association for Computational Linguistics.

Georgie Botev, Arya D. McCarthy, Winston Wu, and David Yarowsky. 2022. Deciphering and characterizing out-of-vocabulary words for morphologically rich languages. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 5309–5326, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

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

Ryan Cotterell, Sabrina J. Mielke, Jason Eisner, and Brian Roark. 2018. Are all languages equally hard to language-model? In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages

536–541, New Orleans, Louisiana. Association for Computational Linguistics.

Mathias Creutz and Krista Lagus. 2002. Unsupervised discovery of morphemes. In *Proceedings of the ACL-02 Workshop on Morphological and Phonological Learning*, pages 21–30. Association for Computational Linguistics.

Vikrant Dewangan, Bharath Raj S, Garvit Suri, and Raghav Sonavane. 2025. When every token counts: Optimal segmentation for low-resource language models. In *Proceedings of the First Workshop on Language Models for Low-Resource Languages*, pages 294–308, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Fitsum Gaim, Wonsuk Yang, and Jong C. Park. 2022. GeezSwitch: Language identification in typologically related low-resourced East African languages. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 6578–6584, Marseille, France. European Language Resources Association.

Leonidas Gee, Leonardo Rigutini, Marco Ernandes, and Andrea Zugarini. 2023. Multi-word tokenization for sequence compression. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 612–621, Singapore. Association for Computational Linguistics.

Daniela Gerz, Ivan Vulić, Edoardo Ponti, Jason Naradowsky, Roi Reichart, and Anna Korhonen. 2018. Language modeling for morphologically rich languages: Character-aware modeling for word-level prediction. *Transactions of the Association for Computational Linguistics*, 6:451–465.

Andargachew Mekonnen Gezmu and Andreas Nürnberger. 2023. Morpheme-based neural machine translation models for low-resource fusion languages. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 22(9):1–19.

Gebrearegawi Gebremariam Gidey, Hailay Kidu Teklehaymanot, and Gebregewergs Mezgebe Atsbha. 2024. Morphological synthesizer for Ge'ez language: Addressing morphological complexity and resource limitations. In *Proceedings of the Fifth Workshop on Resources for African Indigenous Languages @ LREC-COLING 2024*, pages 94–106, Torino, Italia. ELRA and ICCL.

Dan Gillick, Cliff Brunk, Oriol Vinyals, and Amarnag Subramanya. 2016. Multilingual language processing from bytes. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1296–1306, San Diego, California. Association for Computational Linguistics.

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

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

Eylon Gueta, Omer Goldman, and Reut Tsarfaty. 2023. Explicit morphological knowledge improves pretraining of language models for hebrew. *arXiv preprint arXiv:2311.00658*.

Barry Haddow, Rachel Bawden, Antonio Valerio Miceli Barone, Jindřich Helcl, and Alexandra Birch. 2022. Survey of low-resource machine translation. *Computational Linguistics*, 48(3):673–732.

Valentin Hofmann, Hinrich Schuetze, and Janet Pierrehumbert. 2022. An embarrassingly simple method to mitigate undesirable properties of pretrained language model tokenizers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 385–393, Dublin, Ireland. Association for Computational Linguistics.

Jue Hou, Anisia Katinskaia, Anh-Duc Vu, and Roman Yangarber. 2023. Effects of sub-word segmentation on performance of transformer language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7413–7425, Singapore. Association for Computational Linguistics.

Haris Jabbar. 2023. Morphpiece: A linguistic tokenizer for large language models. *arXiv preprint arXiv:2307.07262*.

Marcin Junczys-Dowmunt, Kenneth Heafield, Hieu Hoang, Roman Grundkiewicz, and Anthony Aue. 2018. Marian: Cost-effective high-quality neural machine translation in C++. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 129–135, 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.

Jungseob Lee, Hyeonseok Moon, Seungjun Lee, Chanjun Park, Sugyeong Eo, Hyunwoong Ko, Jaehyung Seo, Seungyoon Lee, and Heuiseok Lim. 2024. Length-aware byte pair encoding for mitigating over-segmentation in Korean machine translation. In

Findings of the Association for Computational Linguistics: ACL 2024, pages 2287–2303, Bangkok, Thailand. Association for Computational Linguistics.

Paul Lerner and François Yvon. 2025. Unlike “likely”, “unlike” is unlikely: BPE-based segmentation hurts morphological derivations in LLMs. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 5181–5190, Abu Dhabi, UAE. Association for Computational Linguistics.

Jindřich Libovický and Jindřich Helcl. 2024. Lexically grounded subword segmentation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 7403–7420, Miami, Florida, USA. Association for Computational Linguistics.

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

Beso Mikaberidze, Temo Saghinadze, Guram Mikaberidze, Raphael Kalandadze, Konstantine Pkhakadze, Josef van Genabith, Simon Ostermann, Lonneke van der Plas, and Philipp Müller. 2024. A comparison of different tokenization methods for the Georgian language. In *Proceedings of the 7th International Conference on Natural Language and Speech Processing (ICNLSP 2024)*, pages 199–208, Trento. Association for Computational Linguistics.

Benjamin Minixhofer, Jonas Pfeiffer, and Ivan Vulić. 2023. CompoundPiece: Evaluating and improving decompounding performance of language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 343–359, Singapore. Association for Computational Linguistics.

Javad Nouri and Roman Yangarber. 2016. A novel evaluation method for morphological segmentation. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC‘16)*, pages 3102–3109, Portorož, Slovenia. European Language Resources Association (ELRA).

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Hyunji Hayley Park, Katherine J. Zhang, Coleman Haley, Kenneth Steimel, Han Liu, and Lane Schwartz. 2021. Morphology matters: A multilingual language modeling analysis. *Transactions of the Association for Computational Linguistics*, 9:261–276.

Maja Popović. 2017. chrF++: words helping character n-grams. In *Proceedings of the Second Conference on Machine Translation*, pages 612–618, Copenhagen, Denmark. Association for Computational Linguistics.

Alfréd Rényi. 1961. On measures of entropy and information. In *Proceedings of the fourth Berkeley symposium on mathematical statistics and probability, volume 1: contributions to the theory of statistics*, volume 4, pages 547–562. University of California Press.

Jonne Saleva and Constantine Lignos. 2021. The effectiveness of morphology-aware segmentation in low-resource neural machine translation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 164–174, Online. Association for Computational Linguistics.

Gabriella F Scelta and Pilar Quezzaire-Belle. 2001. The comparative origin and usage of the ge’ez writing system of ethiopia. *Unpublished manuscript, Boston University, Boston. Retrieved July*, 25:2009.

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.

Hyopil Shin and Hyunjo You. 2009. Hybrid n-gram probability estimation in morphologically rich languages. In *Proceedings of the 23rd Pacific Asia Conference on Language, Information and Computation, Volume 2*, pages 511–520, Hong Kong. City University of Hong Kong.

Michaela Socolof, Jacob Louis Hoover, Richard Futrell, Alessandro Sordoni, and Timothy J. O’Donnell. 2022. Measuring morphological fusion using partial information decomposition. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 44–54, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Xinying Song, Alex Salcianu, Yang Song, Dave Dopson, and Denny Zhou. 2021. Fast WordPiece tokenization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2089–2103, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ahmed Tawfik, Mahitab Emam, Khaled Essam, Robert Nabil, and Hany Hassan. 2019. Morphology-aware word-segmentation in dialectal Arabic adaptation of neural machine translation. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 11–17, Florence, Italy. Association for Computational Linguistics.

Yemane Tedla and Kazuhide Yamamoto. 2018. Morphological segmentation with lstm neural networks for tigrinya. *Int. J. Nat. Lang. Comput*, 7(2):29–44.

Avijit Thawani, Saurabh Ghanekar, Xiaoyuan Zhu, and Jay Pujara. 2023. Learn your tokens: Word-pooled tokenization for language modeling. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9883–9893, Singapore. Association for Computational Linguistics.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *Lrec*, volume 2012, pages 2214–2218. Citeseer.

Thinh Truong, Yulia Otmakhova, Karin Verspoor, Trevor Cohn, and Timothy Baldwin. 2024. Revisiting subword tokenization: A case study on affixal negation in large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5082–5095, Mexico City, Mexico. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Changhan Wang, Kyunghyun Cho, and Jiatao Gu. 2020. Neural machine translation with byte-level subwords. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 9154–9160.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

A Additional Details

Language	**Segmentation Method**	**Size**
Amharic	Morpheme	80k
Amharic	BPE	32k
Tigrinya	Morpheme	80k
Tigrinya	BPE	32k
Bilingual	MoVoC	152k

Table 5: Vocabulary Sizes of BPE and Morpheme-based Vocabularies for Tigrinya and Amharic. The size of the bilingual vocabulary is the sum of all other vocabularies.