

Fast Vocabulary Transfer for African Languages in Multilingual Machine Translation

Stanford CS224N Custom Project

Kailash Elumalai
Department of Computer Science
Stanford University
ckailash@stanford.edu

Biya Brook
Department of Computer Science
Stanford University
bbrook@stanford.edu

Abstract

Multilingual machine translation systems often underperform on low-resource African languages because their shared tokenizers are optimized for high-resource languages and may over-segment morphologically rich or underrepresented text. This leads to a difficult adaptation problem, as adding new vocabulary may improve lexical coverage, but naive expansion can disrupt the pretrained embedding space and hurt translation quality. In this project, we investigate whether *Fast Vocabulary Transfer* (FVT) can make vocabulary expansion more effective when adapting a pretrained NLLB model for English→Amharic and English→Swahili translation. We train language-specific SentencePiece tokenizers on African news text, add previously unseen subword units to the NLLB vocabulary, and compare the baseline model, vocabulary expansion, and FVT initialization. Our results show that tokenizer adaptation has strongly language-dependent effects on translation quality. For Swahili, vocabulary expansion reduces BLEU from 30.01 to 14.99, while FVT partially mitigates this degradation and improves BLEU to 21.67. For Amharic, vocabulary expansion improves BLEU from 8.89 to 10.02, while FVT underperforms both the baseline and vocabulary expansion at 6.96 BLEU. These findings suggest that vocabulary expansion can help languages with severe tokenizer mismatch, while FVT is more useful as a stabilization strategy than as a guaranteed improvement over the pretrained baseline.

Key Information

- Mentor: Julie Kallini
- External Mentor: Tolulope Ogunremi
- External Collaborators: None
- Sharing project: No

1 Introduction

Large multilingual models such as NLLB have dramatically expanded the coverage of machine translation systems, but performance improvements remain uneven across languages. Low-resource African languages in particular often suffer from limited training data and tokenization mismatches between the model’s shared multilingual vocabulary and the linguistic structure of the target language.

No Language Left Behind (NLLB). Our experiments build on the *No Language Left Behind* (NLLB) multilingual translation system introduced by Meta [1]. We use facebook/nllb-200-distilled-600M, a multilingual encoder-decoder model with a shared SentencePiece tokenizer containing 256,000 tokens. Because this tokenizer is trained jointly across

many languages, it can share lexical representations across languages, but it may not represent every language equally well.

This shared-tokenizer design introduces a potential bottleneck for low-resource languages. If a language is underrepresented during tokenizer training, its words may be split into many small subword units rather than represented as coherent lexical pieces. This increases sequence length, weakens lexical representations, and may make translation more difficult. Our work investigates whether adapting the tokenizer vocabulary for specific African languages can alleviate this mismatch.

At the same time, vocabulary adaptation is not straightforward. Adding new language-specific vocabulary may improve lexical coverage, but naive vocabulary expansion also introduces newly initialized embeddings that may be poorly aligned with the pretrained embedding space. In low-resource settings, limited supervision may be insufficient to learn useful representations for these new tokens from scratch.

We investigate whether part of this performance gap can be addressed through vocabulary adaptation. In particular, we explore *Fast Vocabulary Transfer* (FVT) [2], a method for initializing embeddings for newly added tokenizer units by averaging embeddings from the original tokenizer decomposition.

Our work focuses on English→Amharic and English→Swahili translation using NLLB-200 distilled. Amharic uses Ge’ez script and exhibits substantial tokenizer mismatch, while Swahili uses Latin script but still shows moderate fragmentation. Our results show that tokenizer adaptation has strongly language-dependent effects: vocabulary expansion improves performance for Amharic, while for Swahili naive vocabulary expansion hurts performance and FVT partially mitigates this degradation.

Our contributions are:

- A reproducible pipeline for tokenizer adaptation and FVT initialization.
- Quantitative diagnostics measuring tokenizer mismatch for Amharic and Swahili.
- Controlled experiments showing that vocabulary adaptation has strongly language-dependent effects across Amharic and Swahili.

2 Related Work

Large multilingual models such as XLM-R [3] and NLLB [1] demonstrate that large-scale multilingual pretraining enables cross-lingual transfer across hundreds of languages.

Subword tokenization techniques such as BPE [4] and SentencePiece [5] are central to modern NLP pipelines. However, shared tokenizers may fragment words in languages that were underrepresented during tokenizer training.

These lines of work are closely connected. Multilingual models like XLM-R and NLLB rely on shared subword tokenization to transfer across languages, but that same design can disadvantage low-resource languages when the shared vocabulary represents them poorly.

Fast Vocabulary Transfer [2] proposes initializing embeddings for new tokenizer units by averaging embeddings of their original decomposition. While previous work applied FVT in domain adaptation settings, we explore its use for multilingual machine translation in African languages.

This motivates our work as a natural next step. If tokenizer mismatch is one source of poor performance for African languages, then FVT can provide a good way to test whether expanding the vocabulary can improve translation without discarding information from the pretrained model.

Our experiments use the MAFAND-MT dataset [6], which was designed to study adaptation of pre-trained multilingual models to African languages.

3 Approach

Our main method is *Fast Vocabulary Transfer* (FVT) [2], which we apply to multilingual machine translation for African languages. FVT is not a new algorithm, but its application to multilingual machine translation for English→Amharic and English→Swahili translation is the novelty of our approach.

3.1 Baseline model

We use facebook/nllb-200-distilled-600M [1] as our base translation system. This model is a multilingual encoder-decoder translation model with a shared SentencePiece vocabulary of **256,000 tokens**. Our primary baseline is the unmodified NLLB model, which uses the original tokenizer and embedding matrix without any vocabulary expansion. We also compare against vocabulary expansion with randomly initialized new embeddings, which serves as a baseline for evaluating the effect of FVT initialization.

3.2 Fast Vocabulary Transfer

We adopt Fast Vocabulary Transfer (FVT) [2] as our initialization method for newly added tokens. For each new token t that decomposes into $u_1, \dots, u_k$ under the original tokenizer, FVT initializes its embedding as

$$E_{\text{new}}(t) = \frac{1}{k} \sum_{i=1}^k E_{\text{orig}}(u_i).$$

This allows the new tokenizer to represent frequent units directly while preserving pretrained embedding information.

Why FVT may help. Randomly initializing new token embeddings introduces vectors that are initially unrelated to the pretrained embedding space. During fine-tuning, the model must learn useful representations for these tokens from limited training data, which may make optimization more difficult and lead to poorer generalization.

Fast Vocabulary Transfer addresses this issue by initializing new tokens using the embeddings of their original tokenizer decomposition. Because these embeddings already encode semantic information learned during pretraining [3, 2], the resulting vectors begin closer to meaningful regions of the embedding space. In principle, this allows the model to adapt more smoothly when new vocabulary units are introduced.

3.3 Vocabulary Expansion

To improve lexical coverage for the target languages, we construct language-specific vocabulary expansions using target-language text from MAFAND-MT [6]. Our goal is to identify frequent lexical units that are not well represented by the original NLLB tokenizer and add them as new tokens.

We first train SentencePiece BPE tokenizers [5] on target-language text extracted from the MAFAND corpus. Training produces a ranked list of candidate subword units learned directly from the corpus. However, many of these tokens may already appear in the original NLLB tokenizer. To ensure that vocabulary expansion introduces genuinely new lexical units, we compare each candidate token against the existing NLLB vocabulary.

Our token selection pipeline proceeds as follows:

1. Train a SentencePiece tokenizer on the target-language corpus extracted from MAFAND-MT.
2. Extract the learned vocabulary and sort tokens according to the SentencePiece vocabulary order.
3. Remove tokens that already appear in the original NLLB tokenizer vocabulary.
4. Filter punctuation-only tokens and other trivial tokens that do not correspond to meaningful lexical units.
5. Select the top remaining tokens up to a budget of 5,000 tokens.

After overlap filtering with the original NLLB vocabulary, this procedure yields **2,754 new Amharic tokens** and **1,975 new Swahili tokens**. These tokens are appended to the tokenizer vocabulary, and

the model’s embedding matrix is resized accordingly using Hugging Face Transformers. By focusing on tokens that are frequent in the target-language corpus but absent from the original tokenizer, this procedure ensures that vocabulary expansion introduces meaningful lexical units rather than redundant or noisy tokens.

Tokenizer training uses the SentencePiece library [5], and tokenizer/model resizing is implemented with Hugging Face Transformers. The vocabulary-selection pipeline and FVT initialization procedure were implemented by our team.

4 Experiments

4.1 Baseline System

Our baseline system is the pretrained facebook/nllb-200-distilled-600M model used without any modification to its tokenizer or vocabulary. This model uses the original NLLB tokenizer containing **256,000 subword tokens**.

All baseline experiments therefore use the exact vocabulary and embedding matrix provided by the pretrained model. This baseline represents the standard NLLB translation system without tokenizer adaptation.

The vocabulary-expansion and FVT experiments modify this baseline by adding additional tokens to the tokenizer and expanding the embedding matrix accordingly. Importantly, all models share the same architecture, training data, and fine-tuning procedure; the only difference between conditions is how new vocabulary tokens are initialized.

4.2 Datasets

We study English→Amharic and English→Swahili machine translation. We fine-tune on MAFAND-MT [6], where the input is an English sentence and the output is a target-language translation. Evaluation uses FLORES-200 devtest [7, 8], which contains 1,012 examples per language pair.

4.3 Evaluation Method

Our primary evaluation metric is BLEU on FLORES-200 devtest. In addition, we report tokenizer fragmentation statistics, such as the average tokenized sentence length, the median tokenized sentence length, and the 95th percentile tokenized sentence length, to quantify tokenizer mismatch and measure how vocabulary expansion changes tokenization efficiency.

4.4 Tokenizer Diagnostics

We measure tokenization fragmentation on 1,000 FLORES sentences, where fragmentation refers to how many subword tokens the tokenizer produces for a sentence or word.

Language	Avg tokens	Median	95th pct
English	32.3	31	49
Amharic	41.7	39	67
Swahili	36.2	35	57

Table 1: Baseline tokenizer fragmentation statistics.

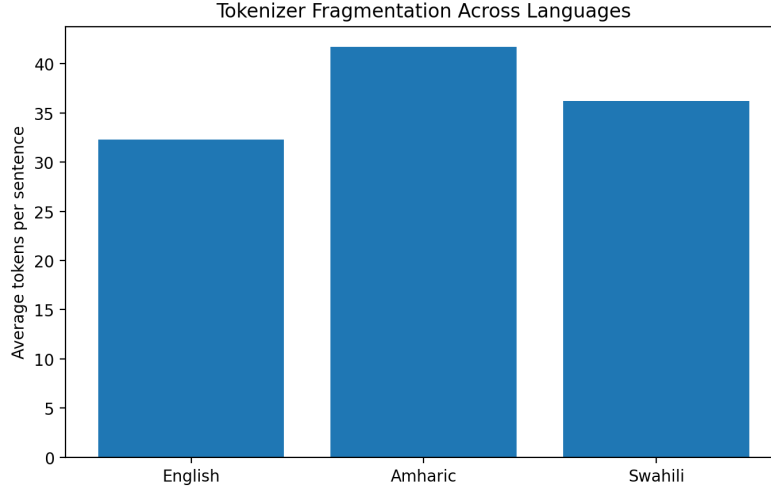


Figure 1: Average tokenized sentence length under the baseline NLLB tokenizer. Amharic and Swahili require more tokens per sentence than English, indicating greater fragmentation.

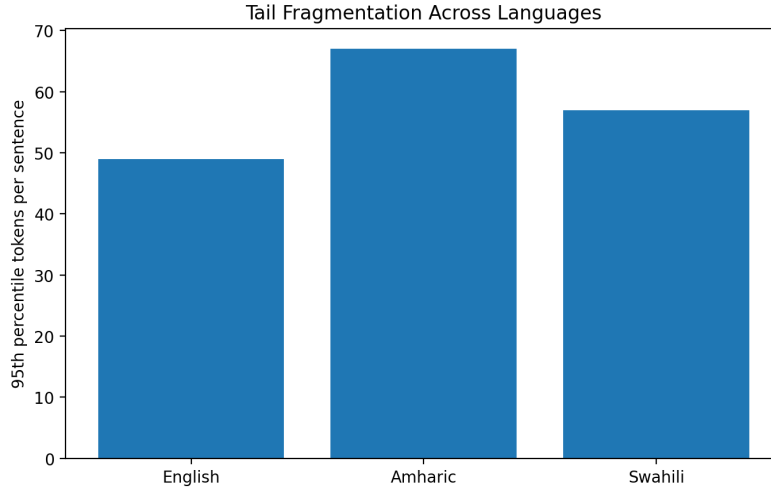


Figure 2: 95th percentile tokenized sentence length under the baseline NLLB tokenizer. The larger tail for Amharic suggests especially severe fragmentation on longer sentences.

4.5 Training setup

All models were fine-tuned using HuggingFace Transformers with:

- Adam optimizer
- learning rate 5×10^{-5}
- batch size 8
- maximum sequence length 128
- beam size 5 during inference

Training was performed on NVIDIA GPUs using Google Colab.

4.6 Implementation Details

All experiments were implemented using the HuggingFace Transformers library. Tokenizer training was performed using the SentencePiece library with byte-pair encoding (BPE). Vocabulary expansion was implemented by modifying the tokenizer configuration and resizing the embedding matrix using the built-in `resize_token_embeddings` function.

For Fast Vocabulary Transfer, new token embeddings were initialized by first decomposing each added token into its original tokenizer representation and then averaging the corresponding embeddings from the pretrained model. The resulting vectors were written directly into the expanded embedding matrix before fine-tuning.

To verify correctness of the implementation, we performed several sanity checks. These included confirming that newly added tokens received consistent token IDs across tokenizer variants, verifying that embedding rows differed between vocabulary expansion and FVT conditions, and manually inspecting several token decompositions.

4.7 Results

Model	Eng→Amh BLEU	Eng→Swh BLEU
Baseline	8.89	30.01
Vocabulary Expansion	10.02	14.99
FVT	6.96	21.67

Table 2: Translation performance on FLORES-200 devtest.

These results show that tokenizer adaptation has strongly language-dependent effects. For Amharic, vocabulary expansion improves BLEU over the baseline, suggesting that improved lexical coverage can help when tokenizer mismatch is severe. For Swahili, however, vocabulary expansion substantially hurts performance, while FVT partially mitigates this drop. At the same time, FVT does not consistently outperform the pretrained baseline, indicating that better initialization alone is not sufficient to guarantee gains from vocabulary adaptation.

Tokenizer	Avg tokens	Median	95th pct
Baseline NLLB	41.7	39	67
Expanded tokenizer	36.4	34	55

Table 3: Tokenization efficiency for Amharic before and after vocabulary expansion.

Vocabulary expansion substantially reduces token fragmentation. For Amharic, the expanded tokenizer reduces average sequence length by approximately 13%. This confirms that many frequent Amharic lexical units were previously split into multiple subword pieces by the baseline tokenizer.

5 Analysis

The results show that vocabulary adaptation has a strong but language-dependent impact.

For Amharic, vocabulary expansion improves BLEU from 8.89 to 10.02, suggesting that improved lexical coverage can benefit languages with severe tokenizer mismatch.

For Swahili, vocabulary expansion initially destabilized training. A rerun with additional training improved expansion to 14.99 BLEU, but FVT substantially outperformed vocabulary expansion at 21.67 BLEU.

5.1 Tokenizer Mismatch

The tokenizer diagnostics reveal a clear mismatch between the baseline tokenizer and African languages. Under the original NLLB tokenizer, Amharic sentences require substantially more tokens than English sentences on average. This fragmentation occurs because many Amharic words are not directly represented in the shared vocabulary.

When words are split into many smaller pieces, the model must compose their meaning across multiple token embeddings. This can make lexical representations less precise and increase the difficulty of translation. In addition, longer token sequences increase computational cost because transformer attention scales quadratically with sequence length.

These observations motivate vocabulary adaptation: if frequent lexical units can be represented as single tokens, the model may be able to represent and generate them more efficiently. Our experiments evaluate whether vocabulary expansion and FVT initialization can help address this mismatch.

These results suggest that embedding initialization plays an important role when introducing new vocabulary units.

5.2 Training Stability

During training we observed that vocabulary expansion occasionally produced unstable outputs containing repeated tokens or malformed sequences. This behavior likely arises because newly added embeddings are randomly initialized and therefore poorly aligned with the pretrained embedding space.

FVT mitigates this problem by initializing new tokens using existing embeddings. Empirically, models using FVT produced more stable translations and converged faster during early training epochs.

5.3 Qualitative examples

Source (English)	Baseline Translation (Swahili)
We now have 4-month-old mice that are non-diabetic that used to be diabetic.	Sasa tuna panya wenye umri wa miezi minne ambao hawana kisukari ambao hapo awali walikuwa na kisukari.

Table 4: Example translation output.

Qualitative inspection of model outputs provides additional insight into the behavior of vocabulary expansion. In several cases, vocabulary expansion produced unstable outputs containing repeated tokens or partially formed words. This behavior likely occurs because randomly initialized embeddings initially have no semantic relationship to the pretrained embedding space.

In contrast, models using Fast Vocabulary Transfer produced more stable and fluent translations. Because FVT initializes new tokens using embeddings derived from existing tokens, the resulting vectors already encode useful semantic information learned during pretraining. This reduces the burden on fine-tuning to learn meaningful representations for new tokens from scratch.

However, qualitative analysis also reveals that the baseline NLLB model often produces strong translations for Swahili. This suggests that tokenizer mismatch may not be the primary bottleneck for languages that already share substantial subword structure with the multilingual tokenizer.

5.4 Why Do Amharic and Swahili Behave Differently?

The differences between the Amharic and Swahili results likely arise from linguistic and tokenization factors. Amharic uses the Ge’ez script, which differs substantially from the Latin scripts used by many high-resource languages. As a result, the shared NLLB tokenizer contains fewer high-quality lexical units for Amharic, leading to greater token fragmentation.

Swahili, by contrast, uses the Latin script and shares more subword structure with high-resource languages that dominate the multilingual tokenizer training corpus. Consequently, the baseline tokenizer already provides reasonable coverage for Swahili vocabulary, which may explain why vocabulary expansion does not improve BLEU relative to the baseline system.

These differences suggest that tokenizer adaptation may be particularly valuable for languages whose scripts or morphology are poorly represented in the original tokenizer training data. In such cases, vocabulary expansion can reduce fragmentation and allow the model to represent common lexical units more directly.

6 Limitations and Future Work

Several limitations should be considered when interpreting our results.

First, our experiments use relatively small fine-tuning datasets from MAFAND-MT. Because vocabulary expansion introduces new parameters in the embedding matrix, larger training datasets may be necessary for the model to fully exploit the additional vocabulary capacity.

Second, we use a relatively small token budget for vocabulary expansion (up to 5,000 tokens). It is possible that larger vocabularies could further reduce token fragmentation for languages such as Amharic. Exploring the relationship between token budget and translation quality would be an interesting direction for future work.

Third, our experiments focus only on English→X translation directions. Investigating bidirectional translation or multilingual fine-tuning could reveal additional effects of tokenizer adaptation.

Finally, while Fast Vocabulary Transfer improves training stability relative to random initialization, it does not consistently outperform the baseline model. Future work could explore alternative initialization strategies, such as weighted averaging of subtokens or contextual embedding initialization.

7 Conclusion

We explored whether Fast Vocabulary Transfer can improve machine translation for African languages. Our results show that tokenizer adaptation significantly affects translation quality but behaves differently across languages. Vocabulary expansion improves Amharic translation quality, while FVT stabilizes training for Swahili relative to vocabulary expansion.

These findings highlight the importance of tokenizer design in multilingual NLP and suggest that vocabulary adaptation remains a promising direction for improving translation in low-resource languages. More broadly, our findings suggest that tokenization may represent an underexplored bottleneck in multilingual NLP, particularly for languages that are typologically distant from those dominating pretraining corpora.

Team Contributions

Kailash implemented the tokenizer adaptation pipeline, including corpus extraction, SentencePiece tokenizer training, vocabulary expansion, and FVT initialization. Kailash also implemented tokenizer diagnostics and baseline evaluation.

Biya implemented the training pipeline, executed the fine-tuning experiments, evaluated model outputs, and contributed to experimental analysis. Biya also worked on organizing and revising the project report.

References

- [1] Marta R. Costa-jussà, James Cross, Onur Celebi, Maha Elbayad, Kenneth Heafield, Kevin Hefernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, et al. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*, 2022.
- [2] Leonidas Gee, Andrea Zugarini, Leonardo Rigutini, and Paolo Torrioni. Fast vocabulary transfer for language model compression. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: Industry Track*. Association for Computational Linguistics, 2022.
- [3] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, 2020.
- [4] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 1715–1725, 2016.

- [5] Taku Kudo and John Richardson. 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, 2018.
- [6] David Adelani, Jesujoba Alabi, Angela Fan, Julia Kreutzer, Xiaoyu Shen, Machel Reid, Dana Ruitter, Dietrich Klakow, Peter Nabende, Ernie Chang, et al. A few thousand translations go a long way! leveraging pre-trained models for african news translation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3053–3070, Seattle, United States, 2022. Association for Computational Linguistics.
- [7] Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc’Aurelio Ranzato, Francisco Guzmán, and Angela Fan. The Flores-101 evaluation benchmark for low-resource and multilingual machine translation. *Transactions of the Association for Computational Linguistics*, 10:522–538, 2022.
- [8] Francisco Guzmán, Peng-Jen Chen, Myle Ott, Juan Pino, Guillaume Lample, Philipp Koehn, Vishrav Chaudhary, and Marc’Aurelio Ranzato. The FLORES evaluation datasets for low-resource machine translation: Nepali–English and Sinhala–English. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6098–6111, Hong Kong, China, November 2019. Association for Computational Linguistics.

A Example Added Tokens

Language	Added Token	Original NLLB Decomposition	English Meaning
Amharic	ኢትዮጵያ	ኢ, ትዮ, ጵያ	Ethiopia
Amharic	አማርኛ	አማ, ር, ኛ	Amharic
Amharic	መንግሥት	መን, ግሥ, ት	government
Swahili	tandao	tanda, o	network
Swahili	blogu	blog, u	blog
Swahili	maendeleo	maende, leo	development

Table 5: Examples of tokens added during vocabulary expansion.

Table 5 shows how these words are decomposed by the original NLLB tokenizer and their English meanings. Many of the added tokens correspond to common lexical units that were previously split into multiple subword pieces. These examples illustrate that the language-specific tokenizer learns meaningful lexical units that correspond to common words or morphemes.