A Case Against Implicit Standards: Homophone Normalization in Machine Translation for Languages that Use the Ge'ez Script

Hellina Hailu Nigatu[1], Atnafu Lambebo Tonja[2], Henok Biadglign Ademtew[3], Hizkel Mitiku Alemayehu[4], Negasi Haile Abadi[5], Tadesse Destaw Belay[6], Seid Muhie Yimam[7]

[1]UC Berkeley, [2] MBZUAI, [3] Vella AI, [4] Paderborn University, [5] Lesan AI, [6]Instituto Politécnico Nacional, [7]University of Hamburg

Correspondence: hellina_nigatu@berkeley.edu

Abstract

Homophone[1] normalization–where characters that have the same sound in a writing script are mapped to one character–is a pre-processing step applied in Amharic Natural Language Processing (NLP) literature. While this may improve performance reported by automatic metrics, it also results in models that are unable to effectively process different forms of writing in a single language. Further, there might be impacts in transfer learning, where models trained on normalized data do not generalize well to other languages. In this paper, we experiment with monolingual training and cross-lingual transfer to understand the impacts of normalization on languages that use the Ge'ez script. We then propose a post-inference intervention in which normalization is applied to model predictions instead of training data. With our simple scheme of post-inference normalization, we show that we can achieve an increase in BLEU score of up to 1.03 while preserving language features in training. Our work contributes to the broader discussion on technology-facilitated language change and calls for more language-aware interventions.

1 Introduction

The majority of the world's languages are underrepresented in natural language processing (NLP) research (Joshi et al., 2020). Collectively, these languages have been referred to as 'low-resource,' owing to the various resources that are not available for them (Nigatu et al., 2024). One of the many resources that are lacking for low-resourced languages is pre-processing tools (Niyongabo et al., 2020). From tokenization methods to basic data cleaning tools, many of the widely used pre-processing schemes do not include, or are not effective for, low-resourced languages (Ahia et al., 2023; Emezue et al., 2023).

Pre-processing steps, ranging from removing punctuation marks to tokenizing text, are essential steps in determining the efficacy of downstream models. For instance, languages that use different writing scripts have been transliterated to a single script to facilitate cross-lingual transfer (Khare et al., 2021). Prior work has explored morpheme-based tokenization for morphologically rich languages as an alternative to word-level tokenization to enhance performance (Tachbelie et al., 2014). Within phonetic languages like Amharic, a common pre-processing intervention has been homophone normalization–i.e, mapping characters with similar sounds to one character (Biadgligne and Smaïli, 2021; Abate et al., 2018).

Homophone normalization has mainly been applied to improve automatic metric scores. Current NLP evaluation schemes, particularly automatic metrics like BLEU (Papineni et al., 2002), which require an exact match between n-grams, do not handle homophone characters. As an example, let us take the homophones <ዓ> and <አ> which both represent the sound /ʔä/ in Amharic. If the Amharic word for "eye" is written as 'ዓይን' in the reference but model prediction outputs 'አይን', evaluation with BLEU score would not count it as a match. However, for an Amharic speaker, those two words have the same pronunciation and meaning. Homophone normalization averts this problem by mapping all homophone characters into a single character and thereby boosting automatic metric scores (Belay et al., 2022). Homophone normalization also reduces the vocabulary size of a dataset, which may be desirable for some applications (Abate et al., 2020). While this indicates a potential benefit in improving performance when using automatic metrics for evaluation, it may lead to downstream issues for language speakers.

In this paper, we argue that the seemingly innocuous act of homophone normalization for Amharic NLP sets and perpetuates an implicit

[1]We use the Merriam-Webster definition of the term homophone: "a character or group of characters pronounced the same as another character or group."

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 10321–10332
November 4-9, 2025

standard for Ge'ez script languages. Currently, homophone normalization is actively being applied to Amharic, one of the many languages that use the Ge'ez script. However, the characters that are normalized in Amharic have distinct sounds in languages such as Tigrinya and Ge'ez. Hence, the implicit standard set by this pre-processing step may have a downstream impact on cross-lingual transfer for the other languages that use the Ge'ez script. Additionally, models trained on normalized datasets will be unable to process alternative word spellings. However, language is not monolithic; normalization may limit how speakers of different dialects and variants of a single language can interact with language technologies. Using Machine Translation (MT) as an NLP task and Amharic, Tigrinya, and Ge'ez as languages of focus, we pose the following research questions:

- **RQ1:** How do existing MT models handle words with homophone characters in languages that use the Ge'ez script?
- **RQ2:** What is the impact of applying different normalization schemes to training data on the performance of MT systems?
- **RQ3:** What is the impact of homophone normalization on transfer learning in MT for related languages?
- **RQ4:** How does applying normalization post-translation compare to applying normalization to the training data?

Multilingual NLP research is often driven by a goal of generalization, proposing ways to make a single model work well for multiple languages (e.g NLLB et al., 2022). While there are demonstrated benefits to this approach, we use our work as a case study to question what we lose through implicit standards in language processing. We find that homophone normalization negatively affects cross-lingual transfer and that applying normalization post-translation boosts automatic scores without compromising language characteristics (Sec. 4). Our work highlights the importance of investigating downstream impacts of preprocessing steps, particularly for low-resourced languages.

2 Background and Related Work

In this section, we provide background on the languages of study and the writing script. We also give background on normalization schemes used in prior work to handle characters with the same sound.

2.1 Languages of Study

The Ge'ez Script is an Abugida writing system– each character in the script represents a consonant and a vowel[2]. Vowels are indicated by modifying the base character. There are 7 vowels in the Ge'ez writing script; hence, each base character has at least 7 variations. For instance, the base character <ለ> /l/ is used to represent the sound /lə/ and is modified to 'ሉ' /lu/, 'ሊ' /li/, and so on. Additionally, there are characters used to represent labiovelars such as 'ኳ'/kwa/. The Ge'ez script is used to write Afro-Semitic languages of Ethiopia and Eritrea, including our languages of focus in this paper: Amharic, Tigrinya, and Ge'ez.

Amharic is an Afro-Semitic language spoken by an estimated 57.5 million people worldwide (Basha et al., 2023). It is primarily spoken in Ethiopia and is one of the federal working languages of the country. The Amharic alphabet has 33 base characters (Adugna).

Tigrinya is an Afro-Semitic language spoken by an estimated 10 million people worldwide (Haile et al., 2023). Tigrinya is one of the federal working languages of Ethiopia and is one of the governmental and national languages of Eritrea. The Tigrinya alphabet has 32 base characters (Negash, 2017).

Ge'ez is an Afro-Semitic language that is currently spoken only as a second language[3]. It is primarily used as a liturgical language within Ethiopian and Eritrean religious institutions. The Ge'ez alphabet has 26 base characters (Demilew, 2019).

2.2 Homophones in the Ge'ez Script

As languages evolve, phonological change occurs where some phonemes might split, merge, or emerge (Boldsen and Paggio, 2022). Since written language evolves at a much slower pace than spoken language, the phonetic changes are usually not reflected in the written forms of language (Obasi, 2018). Due to merged phonemes that are represented by different characters that, in prior years,

[2]https://www.omniglot.com/writing/ethiopic.htm
[3]https://www.ethnologue.com/language/gez/

might have had distinct sounds, the Amharic alphabet has multiple characters that have the same sound (Aklilu, 2010). For instance, all of the following characters in the Ge'ez script <አ>, <ኣ>, <ዐ> or <ዓ> are read as /ʔä/ in Amharic.

Writing scripts are also shared by several languages that may not have evolved in the same way. For the Ge'ez script in particular, some of the characters that have the same sound in Amharic have distinct sounds in Tigrinya. For example, all four characters in the above example that represent /ʔä/ in Amharic have distinct sounds in Tigrinya: <አ> /ʔə/, <ኣ> /ʔä/, <ዐ> /ʕə/ and <ዓ> /ʕä/. There are some characters from the Ge'ez script that have the same sound in Tigrinya, for example 'ሠ' and 'ሰ' both represent /sə/. Due to the differences in how each language uses the characters, altering homophones in the Ge'ez script will have different effects across languages. For example, if you write the word for 'eye', which is written as 'ዓይን ' in Tigrinya as 'አይን', the word would have no meaning, while in Amharic, both words would mean 'eye'. In the Ge'ez language, changing the characters will result in a change in meaning. For instance, the word 'ሰረገ' means 'to hold a wedding' while the word 'ሠረገ' means 'to get inside'.

2.3 Handling Homophone Characters in NLP

Homophone normalization helps improve automatic metric scores by mapping different grapheme variations of a homophone character into a single representation (Sec.1). It has mainly been applied in Amharic NLP literature for Machine Translation (e.g. Abate et al., 2018; Chekole et al., 2024) and semantic modeling tasks (e.g. Belay et al., 2021). However, within papers that report normalizing homophone characters, there is no standard normalization scheme. For instance, some publicly available tools normalize characters with the same sound only (e.g. Kidanemariam, 2019), others normalize characters with the same sound and labialized characters (Mekuriaw and Cohan, 2024; Eshetu, 2022), and some normalize characters with the same sound, labialized characters, and some characters with the same base consonant (Yimam et al., 2021). Further, some prior works report mapping homophone characters to "the most frequently used characters" (Biadgligne and Smaili, 2022; Abate et al., 2018).

While most prior works report using normalization as a standard pre-processing step, Belay et al. (2022) compared MT models trained with and without normalization and reported score improvement for models trained with normalized data. Belay et al. (2021) applied normalization to semantic modeling tasks and found that normalization helped for Information Retrieval but hurt performance for PoS tagging and sentiment analysis. However, these investigations are (1) limited to the Amharic language and (2) do not compare the impact of the different normalization schemes in the literature.

Cases for and against homophone normalization in Amharic: From linguistics literature, there have been three viewpoints on how to handle characters that have the same sound in Amharic: (1) standardize spellings, (2) remove homophone characters from the alphabet–i.e, normalize, or (3) perform no interventions (Aklilu, 2010).

Thus far, the Amharic NLP literature has adopted an (implicit) standardization step with homophone normalization. In this paper, we offer a post-inference intervention that provides a middle ground to the three viewpoints described above. Instead of training on normalized data, we propose performing normalization when calculating a particular metric. We first investigate the impacts of normalization and homophones in MT in zero-shot, monolingual, and cross-lingual settings and show that our post-inference intervention can improve metric scores.

3 Methods

To test the impact of homophone normalization, we prepared an evaluation dataset with a focus on words that have homophone characters in the three languages using publicly available MT datasets (Sec. 3.1). We then adopted two normalization schemes for our experiments, which we describe in Sec. 3.2.

3.1 Dataset

We prepared an evaluation dataset in the three languages of study by focusing on sentences that have high counts of characters with the same sound. In Table 1, we give the details of our dataset[4]. We selected sentences from the following datasets for each language:

[4]Models, code and data can be found at https://github.com/hhnigatu/geez_script_normalization

Target Language	Source Dataset	Training	Eval	Test
Amharic	Abate et al. (2018)	199.2k	22.1k	2.4k
Ge'ez	AGE	15.7k	1.9k	1.9k
Tigrinya	Abate et al. (2018); Lakew et al. (2020)	75.4k	30.1k	2.4k

Table 1: Benchmark dataset description along with source datasets.

Amharic-English-Machine-Translation-Corpus The Amharic-English Machine Translation Corpus (Abate et al., 2018) contains Amharic-English parallel sentences collected from Bible, History, News, and Legal sources. The dataset has a total of 276k parallel sentences. From the test split of the (Abate et al., 2018) dataset, we selected sentences that had at least 9 homophone characters. With this filtering step, we had a test set of 2.4k sentence pairs.

Tigrinya-English MT For Tigrinya, we used data from Lakew et al. (2020) and Abate et al. (2018). The dataset had a total of 150.8k parallel sentences. Similar to Amharic, we selected sentences that had at least 17 homophone characters, which resulted in a test set with 2.4k English-Tigrinya parallel sentences.

AGE We used the AGE dataset (Ademtew and Birbo, 2024) which has 17.5k Amharic-Ge'ez and 18.6k Ge'ez-English parallel sentences. For our experiments, we used the English-Ge'ez data and split it into training, evaluation, and test sets at an 8:1:1 ratio. With this, we had 1.9k Ge'ez-English parallel sentences as our test set. Since the Ge'ez dataset is small, we did not apply additional filtering to the test set.

3.2 Normalization Settings

As discussed in Sec. 2, there are multiple normalization schemes adopted by prior work, particularly when dealing with Amharic datasets. In this study, we employ three normalization settings:

- `No-Norm`: We take the dataset as is, without applying any normalization or other alterations. We use this setting as a baseline.
- `H-only`: We normalize all characters that have the same sound in a given language. We apply this approach for Amharic and Tigrinya, with separate scripts for each language as the characters with the same sound in each language differ (Sec. 2). We map homophone characters to the most frequent character in the dataset.
- `HSL`: In this setting, we use the script from (Yimam et al., 2021) and normalize homophone characters, similar sounds, and labialized characters. Since this approach has only been used for Amharic, and there is no standard way to determine "similar" sounds, we only apply it to the Amharic dataset[5].

Language	No Norm	H-Only	HSL
Amharic	✓	✓	✓
Tigrinya	✓	✓	-
Ge'ez	✓	-	-

Table 2: Application of normalization schemes to the three languages of study.

In Table 2, we give details on how we applied the normalization schemes to our datasets. Note that, for Ge'ez we did not apply any normalization as all characters are distinct–i.e, swapping characters, even if they have the same sound, will result in meaning change (Sec. 2).

4 Experimental Study

In this section, we first give our experimental setup, describing the models we used for our experiments in Sec. 4.1. We conduct experiments on the zero-shot performance of MT systems on sentences with homophone characters (Sec. 4.2). We then investigate the impact of normalizing homophone characters in training data for monolingual model training and cross-lingual transfer (Sec. 4.3). Then, we investigate the efficacy of pots-inference normalization in Sec. 4.4.

4.1 Experimental Setup

4.1.1 Models

Pre-trained MT Models For our zero-shot experiments, we used Google Translate[6], M2M-100-418M (Fan et al., 2021), and NLLB (NLLB et al., 2022) models. All three models support

[5] In (Yimam et al., 2021), characters with 'similar' sounds are some characters that have the same consonant but different vowels; for instance, <ጰ> /t͡s i/ and <ጸ> /t͡s i/ are mapped to <ፀ>. However, there is no standard for determining the similarity of the sounds.

[6] https://translate.google.com/

Amharic, while Google Translate and NLLB support Tigrinya. However, Ge'ez is not included in any of the three models; hence, we did not perform any zero-shot experiment for English-Ge'ez translation.

Models for Training To understand the effects of normalization during training, we (1) fine-tuned the NLLB-600M (NLLB et al., 2022) model and (2) trained an encoder-decoder Transformer model (Vaswani et al., 2017) from scratch. Since the NLLB model includes Amharic and Tigrinya data, we trained the Transformer MT model from scratch to avoid the impact of pre-training in our experimental results.

Training Details We trained an encoder-decoder Transformer model with 8 heads and 6 layers. We used an Adam Optimizer (Kingma and Ba, 2017) with a learning rate of 1e-4 and $\beta 1$ =0.9 and $\beta 2$=0.98. We used a learning rate scheduler that decreased the learning rate by a factor of 0.5 if there were no improvements in 2 consecutive epochs. We used Cross Entropy Loss as our loss function and trained the model for 30 epochs. The best model checkpoint based on evaluation set performance was chosen for the final evaluation. To fine-tune the NLLB-600M (NLLB et al., 2022) model, we used the Trainer module from the HuggingFace transformer library (Wolf et al., 2020). We fine-tuned the model for 5 epochs with a learning rate of 5e-5 using the default training arguments and a batch size of 32. We used the model's default tokenizer without any additional prefixing. We used the same training scheme for all languages and all experiments.

4.1.2 Evaluation

We used both automatic metrics and human evaluation. We performed our evaluation on a single run for each model. For automatic metrics, we used BLEU (Papineni et al., 2002) and ChrF (Popović, 2015). BLEU score focuses on overlap in word-level n-grams, whereas ChrF focuses on character-level n-grams. We used the SACREBLEU (Post, 2018) implementation for both BLEU and ChrF, with their default settings. When calculating the scores, we removed punctuation marks from both reference and prediction sentences. For all test cases, we apply the same normalization scheme to the reference and prediction. This makes it difficult to make comparisons across the references; for instance, we cannot directly compare the reference without normalization to the reference with H-only normalization. However, as described in Sec. 2, the motivation for applying normalization is to increase scores of automatic metrics by "standardizing" the spelling of words in a given language with a normalization scheme. Further, depending on the language it is applied to, normalization affects the spelling of a word and not the meaning. Hence, references with different normalization settings applied carry the same semantic meaning, although they may differ in the spelling of some words.

For human evaluation, native speakers of Tigrinya and Amharic and second language speakers of Ge'ez qualitatively looked at 50 random sample predictions, comparing the outputs of the different models. We focused on the following axes when evaluating: (1) rating which translation was better from the given models, (2) identifying words that were mistranslated (e.g, words that were in Amharic for Tigrinya translations and vice versa), and (3) identifying changes in homophones in the translations.

Model	Amharic		Tigrinya	
	BLEU	ChrF	BLEU	ChrF
NLLB - 3B	10.47	34.05	11.26	31.22
NLLB - 600M	6.98	29.16	11.55	31.30
Google Translate	9.89	33.67	16.02	38.75
M2M - 418M	13.51	34.78	-	-

Table 3: Zero-Shot translation performance.

4.2 Zero-Shot Experiments

This experiment aims to answer **RQ1**–that is, to understand if there is an existing impact on pre-trained MT models in handling characters with the same sound in the three languages of study.

Results As can be seen in Table 3, all models except the NLLB-200-Distilled-600M have comparable ChrF scores, with M2M-100 having the highest ChrF for Amharic. For Tigrinya, Google Translate had the highest BLEU and ChrF scores. Additionally, M2M-100 has the highest BLEU score for Amharic, while NLLB-200-Distilled-600M had the lowest BLEU score. Further, the open-sourced NLLB-200-Distilled-600M and M2M-100 models performed better than the commercially available Google Translate model for Amharic.

Qualitatively, we observe that the outputs of the NLLB models for English-Amharic translation usually stick with the Amharic "Standard"

Language	Stage	Model	No Norm		H-only		HSL	
			BLEU	ChrF	BLEU	ChrF	BLEU	ChrF
Tigrinya	Training	Transformer	**10.87**	25.51	10.21	**26.61**	-	-
		NLLB	**22.71**	**43.11**	21.41	41.78	-	-
Amharic	Training	Transformer	**12.32**	**29.50**	9.31	26.90	6.22	26.88
		NLLB	19.09	41.98	**19.71**	**42.59**	17.13	40.50
	Inference	Transformer	12.32	29.50	**12.56**	29.77	**12.56**	**29.79**
		NLLB	19.09	41.98	**19.78**	42.60	**19.78**	**42.61**
		Belay et al. (2022)	13.51	34.78	**14.54**	35.94	**14.54**	**35.95**

Table 4: Performance of models where normalization is applied during `Training` and `Inference`. Best performance for each row is indicated in bold.

homophone usage (Sec. 2). This behavior is not observed in Google Translate. For instance, when translating the word "God," all NLLB and M2M models consistently translate it as "እግዚአ-ብሔር", which is consistent with the Ge'ez spelling of the word. However, Google Translate sometimes tends to translate it as "እግዚአብሄር", switching the <ሐ> character with <ሄ> which does not conform to the standard homophone usage of the word (Aklilu, 2010). This difference between the open-source models and Google Translate may be due to the fact that open models are trained on publicly available data, which is heavily dominated by religious data for low-resourced languages.

4.3 Effects of Training Models with Homophone Normalization

To answer **RQ2** and **RQ3**, we trained an encoder-decoder Transformer model from scratch and fine-tuned an NLLB-600M model as described in Sec. 4.1. We experimented with monolingual and cross-lingual training, which we describe below.

4.3.1 Monolingual Effects of Normalization

For **RQ2**, we experimented by training a Transformer model from scratch and finetuning NLLB-600M for each of the three language pairs: Eng-Amh, Eng-Tir, and Eng-Ge'ez. The goal for this experiment was to understand the impact of normalizing homophone characters in the target language during training on the MT performance. As described in Sec. 3, we use the `No-Norm` setting as a baseline for all languages, apply `H-Only` normalization to Amharic and Tigrinya data, apply `HSL` normalization to Amharic data only. For Ge'ez, we train without any normalization (Sec. 2).

Results As can be seen in Table 4, for both Amharic and Tigrinya, when normalization is applied during training, the model with `No-Norm` has better BLEU score as compared to the models trained on normalized data for the Transformer models. For Tigrinya, we observe that the Transformer model has comparable performance with and without normalization. For NLLB-600M, `H-Only` has a marginal improvement over the `No-Norm` setting for Amharic (+0.62 BLEU and +0.61 ChrF). The `HSL` setting has the least performance with NLLB for Amharic. For Tigrinya, we observe that `No-Norm` has better performance than the `H-Only` setting for the fine-tuned NLLB model (+1.3 BLEU and +1.33 ChrF).

Qualitatively, we observed that models trained with `HSL` normalization mostly replace some words with their synonyms and simplify the translation when compared to `H-Only` and `No-Norm` settings. Regarding the quality of the translation, NLLB fine-tuned with `No-Norm` setting provides better translation, preserving the homophone characters in the prediction; this also aligns with the automatic result presented in Table 4. In the `H-Only` setting, we noticed that in addition to replacing words with normalized characters, most of the translations were incomplete even though we set the same maximum sequence length for all models.

4.3.2 Cross-Lingual Transfer Effects of Normalization

For **RQ3**, we experimented by taking the models we trained for Amharic as described in Sec 4.3.1 and further training with Eng-Tir and Eng-Ge'ez data. In our cross-lingual experiment, the Tigrinya and Ge'ez datasets are taken as is, without any normalization.

Results As Table 5 shows, for Tigrinya, we find that the Amharic model that is trained without normalizing the homophone characters–i.e. the `No-Norm` setting–is a better transfer model

Tigrinya

	No-Transfer		No-Norm		H-only		HSL	
Model	BLEU	ChrF	BLEU	ChrF	BLEU	ChrF	BLEU	ChrF
Transformer	10.87	25.51	**12.16**	**27.61**	10.67	26.24	11.23	26.44
NLLB-600M	**22.71**	**43.11**	21.55	42.03	21.63	42.13	21.68	42.14

Ge'ez

	No-Transfer		No-Norm		H-only		HSL	
Model	BLEU	ChrF	BLEU	ChrF	BLEU	ChrF	BLEU	ChrF
Transformer	2.46	18.72	**3.67**	**20.80**	3.56	**20.80**	1.46	12.48
NLLB-600M	3.36	23.48	5.22	26.54	**6.33**	**28.38**	6.31	28.52

Table 5: Performance of MT models in cross-lingual transfer experiments, where `No-Norm`, `H-Only`, and `HSL` refer to models that were initialized with English-Amharic models trained in each of the three settings. The best performance in a row is indicated in bold font.

as compared to the `H-Only` and `HSL` settings for the Transformer models. With finetuning NLLB-200-Distilled-600M, we find that the model directly finetuned from NLLB-200-Distilled-600M performed better than the ones first finetuned on Amharic then finetuned on Tigrinya. With the transfer models for NLLB, we observe comparable performance regardless of the normalization setting with which the Amharic model was finetuned. For Ge'ez, we see that using the Amharic models trained in the `No-Norm` and `H-Only` setting provides better BLUE and ChrF scores as compared to using the model trained with `HSL` setting. Further, we observe that using a Transformer model that was trained with the `HSL` setting for Amharic has the worst performance when used as a transfer model for Ge'ez. Qualitatively, we observe that in both Ge'ez and Tigrinya translations, the output includes code-switching with Amharic words, changes pronouns, changes gender, and wrongly negates words.

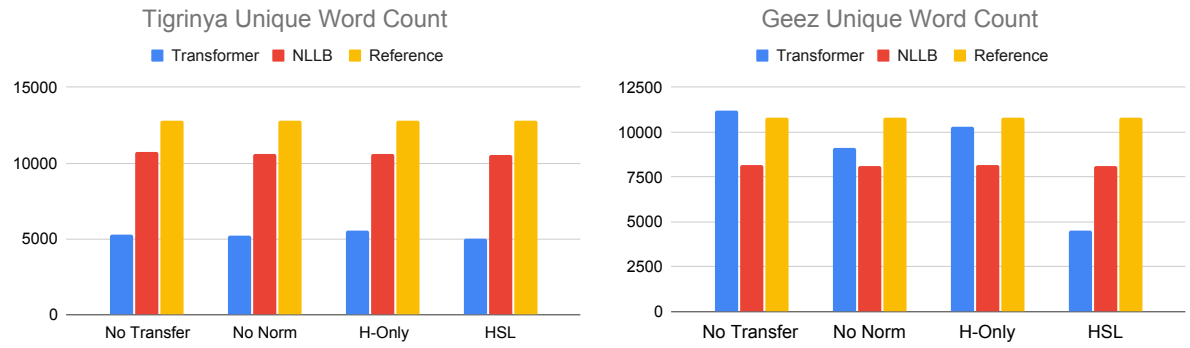
We find that the homophone characters that were normalized in the Amharic transfer model were correctly used in the respective target languages (Tigrinya and Ge'ez). However, looking at the predictions of the models fine-tuned on the transfer models trained using the Amharic normalized data, they tended to repeat characters or words until they reached the maximum sequence length, instead of translating the source sentence. This is demonstrated in Figure 1, where the translations with the models trained on Amharic transfer models with the `HSL` setting have fewer unique words in both Ge'ez and Tigrinya, especially for the Transformer model. In Table 6, we provide qualitative examples.

Looking at the character count of the translations in the different transfer settings, we find that all models did not contain a comparable number of characters that were found in the reference dataset; for instance, for Ge'ez the model trained without transfer learning did not contain 33 characters that were in the reference dateset while the model trained with the `HSL` normalized Amharic transfer model did not contain 34 characters form the reference. However, training with the Amharic transfer models added new characters in the predictions, where the characters do not exist in the language. For instance, <ሸ>, <ሽ>, <ቸ> were added in the Ge'ez predictions although all three characters do not exist in the alphabet for the language.

4.4 Post-Inference Normalization

As discussed in Sec. 2, normalization of homophones has provided automatic score increases in prior work. However, normalizing the characters before training a model results in models that cannot process different forms of spelling. Furthermore, as we have seen in Sec. 4.3.2, normalizing homophone characters has an impact on transfer learning for languages that use the same writing script. To answer our **RQ4**, we took the models we trained in the `No-Norm` setting and applied normalization to the reference and predictions after inference.

Results For the Transformer model we trained, post-normalization improves BLEU and ChrF scores by a small margin (0.24 and 0.29 increase, respectively). For the NLLB finetuned model, we find that applying `HSL` normalization post-inference boosts the BLEU score by 0.69 and the ChrF by 0.63. In the three normalization settings,



(a) (b)

Figure 1: Comparison of Unique word count with different transfer settings for English-Tigrinya and English-Ge'ez translation.

we find that the NLLB model outperforms the Transformer model on our evaluation dataset.

We compare how effective post-inference normalization is by including the model from Belay et al. (2022); we take the model trained without normalization and apply homophone normalization after inference[7]. Belay et al. (2022) found a 3.09 BLEU score increase by finetuning an M2M (Fan et al., 2021) model with `HSL` normalized data as compared to a model trained with `No-Norm` data. We cannot directly compare our results with the reported BLEU scores as the test sets are different. However, on our evaluation dataset, we find that the model trained by Belay et al. (2022) without applying normalization can have a 1.03 BLEU score increase (Table 4) with our post-inference scheme.

5 Discussion

Our work investigates the impact of homophone normalization for languages that use the Ge'ez script on Machine Translation performance. We provide background on the characteristics of the languages that use the Ge'ez script and detail how prior work used homophone normalization (Sec. 2). Through a series of experiments (Sec. 4), we demonstrate that homophone normalization does not provide a significant performance gain across all languages, and hurts performance in transfer learning (Sec. 4.3). As we have discussed in Sec. 2, homophone normalization has been used as a pre-processing step in the NLP literature for Amharic, setting an implicit standard on what trained models can handle. In this section, we connect this argument to the broader literature on technology-facilitated language change.

Evolutions in language that are the result of technological constraints make their way to daily lives (van Dijk et al., 2016). This is particularly concerning as MT models are used in data creation and augmentation for low-resourced languages (e.g Singh et al., 2025). Machine-translated datasets are also used to train other NLP models (e.g Joshi et al., 2025), perpetuating the normalization effect to tasks beyond translation.

As we have seen in Sec. 4, while normalization has resulted in score improvements in prior work, it affects the performance of models in transfer learning. Further, the score improvements are not consistent across models, languages, and normalization settings. As a result, we need to pause and reflect on using such schemes in NLP literature for languages that use the Ge'ez script. Multiple languages use the same writing script; hence, it is important to consider how the standards we set for one language affect other languages. There might also be dialect differences in how words are spelled, which will not be accounted for when we normalize homophone characters without such considerations.

As the number of low-resourced languages represented in NLP research increases, it is imperative to consider how pre-processing steps applied to these languages alter the overall landscape of language use. Design decisions could lead to constraints on how and if people can use their language (Wenzel and Kaufman, 2024). In the con-

[7]We could not compare with the models trained with normalization from Belay et al. (2022) as they are not publicly available.

text of our study, training models on normalized data results in models that cannot handle alternative spellings. For instance, (Belay et al., 2021) found that normalization helped improve performance in information retrieval. However, the performance improvement would require users to conform to the normalized form of spelling. This impact is not limited to homophone normalization; Adebara and Abdul-Mageed (2022) argue that normalizing tone diacritics, which are essential for lexical disambiguity, affects the usability of retrieval systems for African language speakers.

Further, relying solely on automatic score improvements obfuscates the impact of our design decisions beyond its intended effect. Instead, our solutions should (**1**) focus on changing the methods (e.g. the metrics used for evaluation), (**2**) be explicit under what context the improvements are achieved, and (**3**) explore alternatives that do not impact the model's ability to handle different versions of a language. As we propose in Sec. 4.4, we can use post-inference interventions to increase automatic scores without altering the training data. Since there are no standard ways of spelling agreed upon and people spell words differently, with the post-inference normalization, researchers can see to what degree the performance they are getting is a result of spelling differences due to homophones vs actual issues with the translation model, without limiting the inputs and outputs of their models. While the performance improvement is not as significant as training on normalized data, it is a tradeoff for having a model that can account for different spellings, dialects, and transfer capabilities.

6 Conclusion

We investigated the impact of homophone normalization on languages that use the Ge'ez script. We find that normalization of homophones in training data leads to poor transfer learning performance for related languages. Furthermore, we find that normalization does not always lead to performance improvement across all languages. We argue against implicit standardization via preprocessing tools and offer an alternative approach that preserves features of the languages during training. We use our work as a case study to call for a more thorough examination of preprocessing steps, particularly for low-resource languages.

Limitations

While our experiments show an increase in the BLEU score with post-inference homophone normalization, we did not conduct a full-scale human evaluation of translation quality; instead, we manually inspected 50 outputs across all normalization settings. Future work should include large-scale human evaluations. Our conclusion is mainly based on BLEU and ChrF scores, while they remain standard evaluation tools for MT, they might not show the changes in prediction when we use different normalization settings.

Acknowledgments

We thank the reviewers for their valuable feedback. We also extend our gratitude to Nina Markl for providing feedback on our manuscript.

References

Solomon Teferra Abate, Michael Melese, Martha Yifiru Tachbelie, Million Meshesha, Solomon Atinafu, Wondwossen Mulugeta, Yaregal Assibie, Hafte Abera, Binyam Ephrem, Tewodros Abebe, Wondimagegnhue Tsegaye, Amanuel Lemma, Tsegaye Andargie, and Seifedin Shifaw. 2018. Parallel Corpora for bi-lingual English-Ethiopian Languages Statistical Machine Translation.

Solomon Teferra Abate, Martha Yifiru Tachbelie, and Tanja Schultz. 2020. Multilingual Acoustic and Language Modeling for Ethio-Semitic Languages. In *Interspeech 2020*, pages 1047–1051. ISCA.

Ife Adebara and Muhammad Abdul-Mageed. 2022. Towards afrocentric NLP for African languages: Where we are and where we can go. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3814–3841, Dublin, Ireland. Association for Computational Linguistics.

Henok Ademtew and Mikiyas Birbo. 2024. AGE: Amharic, Geez and English Parallel Dataset. In *Proceedings of the Seventh Workshop on Technologies for Machine Translation of Low-Resource Languages (LoResMT 2024)*, pages 139–145, Bangkok, Thailand. Association for Computational Linguistics.

Gabe Adugna. Research: Language Learning - Amharic: Home.

Orevaoghene Ahia, Sachin Kumar, Hila Gonen, Jungo Kasai, David Mortensen, Noah Smith, and Yulia Tsvetkov. 2023. Do All Languages Cost the Same? Tokenization in the Era of Commercial Language Models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*,

pages 9904–9923, Singapore. Association for Computational Linguistics.

Amsalu Aklilu. 2010. *Problems of Writing Homophones without care and its Solution [Title translated from Amhairc]*.

Shaik Johny Basha, Duggineni Veeraiah, Boddu Venkat Charan, Wiltrud Sahithi Joyce Yeddu, and Devalla Ganesh Babu. 2023. Detection and Comparative Analysis of Handwritten Words of Amharic Language to English using CNN-Based Frameworks. In *2023 International Conference on Inventive Computation Technologies (ICICT)*, pages 422–427. ISSN: 2767-7788.

Tadesse Destaw Belay, Abinew Ali Ayele, Getie Gelaye, Seid Muhie Yimam, and Chris Biemann. 2021. Impacts of Homophone Normalization on Semantic Models for Amharic. In *2021 International Conference on Information and Communication Technology for Development for Africa (ICT4DA)*, pages 101–106.

Tadesse Destaw Belay, Atnafu Lambebo Tonja, Olga Kolesnikova, Seid Muhie Yimam, Abinew Ali Ayele, Silesh Bogale Haile, Grigori Sidorov, and Alexander Gelbukh. 2022. The Effect of Normalization for Bi-directional Amharic-English Neural Machine Translation. *arXiv preprint*. ArXiv:2210.15224 [cs].

Yohanens Biadgligne and Kamel Smaïli. 2021. Parallel Corpora Preparation for English-Amharic Machine Translation. In *Advances in Computational Intelligence*, pages 443–455. Springer, Cham. ISSN: 1611-3349.

Yohannes Biadgligne and Kamel Smaili. 2022. Offline Corpus Augmentation for English-Amharic Machine Translation. In *2022 5th International Conference on Information and Computer Technologies (ICICT)*, pages 128–135.

Sidsel Boldsen and Patrizia Paggio. 2022. Letters from the past: Modeling historical sound change through diachronic character embeddings. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6713–6722, Dublin, Ireland. Association for Computational Linguistics.

Adane Kasie Chekole, Tesfa Tegegne Asfaw, Tesfahun Nurrie Mengestie, Belayneh Teshome Kebie, Mengistu Kinfe Negia, and Yohannes Abinet Worku. 2024. Effect of Parallel Data Processing Model on Bi-Directional English-Khimtagne Machine Translation Using Deep Learning. In *2024 International Conference on Information and Communication Technology for Development for Africa (ICT4DA)*, pages 189–193.

Fitehalew Ashagrie Demilew. 2019. ANCIENT GEEZ SCRIPT RECOGNITION USING DEEP CONVOLUTIONAL NEURAL NETWORK. *Software Engineering*.

Chris Emezue, Hellina Nigatu, Cynthia Thinwa, Helper Zhou, Shamsuddeen Muhammad, Lerato Louis, Idris Abdulmumin, Samuel Oyerinde, Benjamin Ajibade, Olanrewaju Samuel, Oviawe Joshua, Emeka Onwuegbuzia, Handel Emezue, Ifeoluwatayo A. Ige, Atnafu Lambebo Tonja, Chiamaka Chukwuneke, Bonaventure F. P. Dossou, Naome A. Etori, Mbonu Chinedu Emmanuel, Oreen Yousuf, Kaosarat Aina, and Davis David. 2023. The African Stopwords project: curating stopwords for African languages. *arXiv preprint*. ArXiv:2304.12155 [cs].

Abebawu Eshetu. 2022. Amharic-Simple-Text-Preprocessing-Usin-Python. Original-date: 2019-08-05T09:30:04Z.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Edouard Grave, Michael Auli, and Armand Joulin. 2021. Beyond english-centric multilingual machine translation. *J. Mach. Learn. Res.*, 22(1):107:4839–107:4886.

Negasi Haile, Nuredin Ali, and Asmelash Teka Hadgu. 2023. ERROR ANALYSIS OF TIGRINYA ENGLISH MACHINE TRANSLATION SYSTEMS.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The State and Fate of Linguistic Diversity and Inclusion in the NLP World. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, Online. Association for Computational Linguistics.

Raviraj Joshi, Kanishk Singla, Anusha Kamath, Raunak Kalani, Rakesh Paul, Utkarsh Vaidya, Sanjay Singh Chauhan, Niranjan Wartikar, and Eileen Long. 2025. Adapting Multilingual LLMs to Low-Resource Languages using Continued Pretraining and Synthetic Corpus. *arXiv preprint*. ArXiv:2410.14815 [cs].

Shreya Khare, Ashish Mittal, Anuj Diwan, Sunita Sarawagi, Preethi Jyothi, and Samarth Bharadwaj. 2021. Low Resource ASR: The Surprising Effectiveness of High Resource Transliteration. In *Interspeech 2021*, pages 1529–1533. ISCA.

Bushra Kidanemariam. 2019. Amharic-NLP-Tools-in-JAVA.

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A Method for Stochastic Optimization. *arXiv preprint*. ArXiv:1412.6980 [cs].

Surafel M. Lakew, Matteo Negri, and Marco Turchi. 2020. Low Resource Neural Machine Translation: A Benchmark for Five African Languages. *arXiv preprint*. ArXiv:2003.14402 [cs].

Daniel Mekuriaw and Arman Cohan. 2024. BENCHMARK DATASET AND PARAMETER-EFFICIENT CROSS-LINGUAL TRANSFER LEARNING FOR AMHARIC TEXT SUMMARIZATION. Technical report.

Merriam-Webster. Definition of HOMOPHONE.

Abraham Negash. 2017. The Origin and Development of Tigrinya Language Publications (1886 ...

Hellina Hailu Nigatu, Atnafu Lambebo Tonja, Benjamin Rosman, Thamar Solorio, and Monojit Choudhury. 2024. The Zenos Paradox of Low-Resource Languages. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17753–17774, Miami, Florida, USA. Association for Computational Linguistics.

Rubungo Andre Niyongabo, Qu Hong, Julia Kreutzer, and Li Huang. 2020. KINNEWS and KIRNEWS: Benchmarking cross-lingual text classification for Kinyarwanda and Kirundi. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5507–5521, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Team NLLB, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No Language Left Behind: Scaling Human-Centered Machine Translation. *arXiv preprint*. ArXiv:2207.04672 [cs].

Jane Chinelo Obasi. 2018. Structural Irregularities within the English Language: Implications for Teaching and Learning in Second Language Situations.

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, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Shivalika Singh, Angelika Romanou, Clémentine Fourrier, David I. Adelani, Jian Gang Ngui, Daniel Vila-Suero, Peerat Limkonchotiwat, Kelly Marchisio, Wei Qi Leong, Yosephine Susanto, Raymond Ng, Shayne Longpre, Wei-Yin Ko, Sebastian Ruder, Madeline Smith, Antoine Bosselut, Alice Oh, Andre F. T. Martins, Leshem Choshen, Daphne Ippolito, Enzo Ferrante, Marzieh Fadaee, Beyza Ermis, and Sara Hooker. 2025. Global MMLU: Understanding and Addressing Cultural and Linguistic Biases in Multilingual Evaluation. *arXiv preprint*. ArXiv:2412.03304 [cs].

Martha Yifiru Tachbelie, Solomon Teferra Abate, and Laurent Besacier. 2014. Using different acoustic, lexical and language modeling units for ASR of an under-resourced language Amharic. *Speech Communication*, 56:181–194.

Chantal N. van Dijk, Merel van Witteloostuijn, Nada Vasi, Sergey Avrutin, and Elma Blom. 2016. The Influence of Texting Language on Grammar and Executive Functions in Primary School Children. *PLoS ONE*, 11(3):e0152409.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need.

Kimi Wenzel and Geoff Kaufman. 2024. Designing for Harm Reduction: Communication Repair for Multicultural Users' Voice Interactions. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, pages 1–17, Honolulu HI USA. ACM.

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.

Seid Muhie Yimam, Abinew Ali Ayele, Gopalakrishnan Venkatesh, Ibrahim Gashaw, and Chris Biemann. 2021. Introducing Various Semantic Models for Amharic: Experimentation and Evaluation with Multiple Tasks and Datasets. *Future Internet*, 13(11):275. Number: 11 Publisher: Multidisciplinary Digital Publishing Institute.

Target Lang.	Source	Reference	No-Norm	H-Only	HSL
Tir	The discourse will also answer such questions as these : How often should Christians commemorate this event ?	ብተወሳኺ እቲ ዝርርብ ንኽምዚ ዚስዕብ ዝኣመሰለ ሕቶታት መልሲ ኺህብ እዩ - ክርስትያናት ነዛ ዕለት እዚኣ ኼብ-ዕልዋ ዘለዎም ክንደይ ሳዕ እዩ ፧	ብዘይካዚ ፡ ነዚ ዚስዕብ ሕቶታት መልሲ ይህብ ፦ ክርስትያናት ፡ ነዚ ዚስዕብ ሕቶታት እዚ ፡ ኣብ ላዕሊ ዘሎ መልሲ ኺውሃሪ ፦ መብዛሕትኡ እዋን ፡ ክርስትያናት ነዚ ፍጻመ እዚ ብኽመይ ኪምልስ ይግባእ፧	እዚ ነዚ ነዚ ነዚ ፍጻመ እዚ ፍጻመ እዚ ከምዚ ዝ ክርስትያናት ክሳዕ ዜ ከምዚ ዝ ፦ ብኽመይ ኪ ክርስትያናት ክሳዕ ነዚ ነዚ ነዚ ነዚ ነዚ ነዚ ነዚ ነዚ ነዚ ፍጻመ እዚ ፍጻመ እዚ ፍጻመ እዚ ብኽመይ ክ ኼ ብኽመይ ክ[...]	እቲ ተባሂሉ ዚብል ዚብል ዚብል ሕቶታት ዚብል ሕቶታት ዚብል ሕቶታት ዚብል ዚብል ዚብል ዚብል ዚብል ዚብል ዚብል ዚብል ዚብል ዚብል ዚብል ዚብል ዚብል ዚብል ዚብል ዚብል[...]
Ge’ez	For this reason a man and a woman are to become one, and for this reason he leaves his father and his mother. He associates with his wife, and they become one flesh.	በእንተዝ ይኩን ብእሲ ወብእሲት አሐደ ወበእንተዝ የኀድግ አባሁ ወእሞ ወይፃመር ምስለ ብእሲቱ ወይከውኑ ሥጋ አሐደ።	ወበእንተ ዝንቱ ይኅድግ ብእሲ አባሁ ወእሞ ዝ ፈለግ ወይ-ትልዋ ለብእሲቱ ወይከውኑ ክል-ኤሆሙ አሐደ ሥጋ ወአሐደ እምአእመት።	በእንተዝ ብእሲ አሐዱ ብእሲ ምስለ ካልኡ ወአደመሙ አውዶሙ ወያ-ገብኦ አባሁ ወእሞ ወይተልዋ ለብ-እሲቱ ወይከውኑ ክልኤሆሙ አሐደ ሥጋ።ወስብሐት ለእግዚአብሔር አሜን።	ወትቤ ዝንቱ ውእቱ ብእሲ ወብእሲ እመቦ ከመ ብእሲ አባሁ ወእሞ ወብእሲ እሙ ወአኀዊሁ ኅቡረ ብእሲቱ ሥጋ ሥጋ ሥጋ ሥጋ ሥጋ ሥጋ ሥጋ ሥጋ ሥጋ ሥጋ ሥጋ ሥጋ ሥጋ ሥጋ ሥጋ ሥጋ ሥጋ ሥጋ ሥጋ አሐደ ሥጋ።

Table 6: Qualitative examples with transfer learning experiments where the transfer Amh-Eng model is trained in No-Norm, H-Only and HSL settings

A Qualitative Examples

In Table 6, we provide qualitative examples for Tigrinya and Ge’ez transfer learning experiments. As the table shows, Amharic models trained with normalization repeat words until they reach maximum sequence length or end of sentence token (።).