Impacts of Homophone Normalization on Semantic Models for Amharic

Tadesse Destaw Belay
College of Informatics
Wollo University
Kombolcha, Ethiopia
tadesseit@gmail.com

Abinew Ali Ayele
ICT4D Research Center
Bahir Dar University
Bahir dar, Ethiopia
abinewaliayele@gmail.com

Getie Gelaye
African Linguistics & Ethiopian Studies
Universität Hamburg
Hamburg, Germany
getie.gelaye@uni-hamburg.de

Seid Muhie Yimam
Dept. of Informatics
Universität Hamburg
Hamburg, Germany
seid.muhie.yimam@uni-hamburg.de

Chris Biemann
Dept. of Informatics
Universität Hamburg
Hamburg, Germany
christian.biemann@uni-hamburg.de

Abstract*—Amharic is the second-most spoken Semitic language after Arabic and serves as the official working language of the government of Ethiopia. In Amharic writing, there are different characters with the same sound, which are called homophones. The current trend in Amharic NLP research is to normalize homophones into a single representation. This means, instead of character ሀ(hä[1]), ሃ(ha), ሐ(ḥä), ሓ(ḥa), ኀ(ḫä), ኃ(ḫa), and ኸ(ḫa), the character ሀ(hä) will be used; instead of አ(a), ኣ(ʔ), ዐ(ʔä), and ዓ(ʔa), the character አ(a) will be replaced; and so on. This was done by the assumption that they are repetitive alphabets as they have the same sound. However, the impact of homophone normalization for Amharic NLP applications is not well studied. When one homophone character is substituted by another, there will be a meaning change and it is against the Amharic writing regulation. For example, the word ድህነት (dəhənatə) is "poverty" while ድኅነት (dəḫənatə) means "salvage". These two words are homophones, but they have different meanings. To study the impacts of homophone normalization, we develop different general-purpose pre-trained embedding models for Amharic using regular and normalized homophone characters. We fine-tune the pre-trained models and build some Amharic NLP applications. For PoS tagging, a model that employs a regular FLAIR embedding model performs better, achieving an F1-score of 77%. For sentiment analysis, the model from regular RoBERTa embedding outperforms the other models with an F1-score of 60%. For IR systems, we achieve an F1-score of 90% using the normalized document. The results show that normalization is highly dependent on the NLP applications. For sentiment analysis and PoS tagging, normalization has negative impacts while it is essential for IR. Our research indicates that normalization should be applied with caution and more effort towards standardization should be given.

Keywords— homophone normalization, NLP applications, semantic models, pre-trained models

I. Introduction

Amharic is the second-most spoken Semitic language after Arabic and serves as the official working language of the government of Ethiopia and many regional states in the country [1, 2]. In Amharic writing, there are different characters with the same sound but different in shape and meaning, which are called homophones (ድምጸ ሞክሼ ሆሄያት – dəməṣa mokəshe hoheyatə). For instance, ሀ(hä), ሐ(ḥa), ኀ(ḫä), and ኸ(ḫe); አ(a) and ዐ(ʔä); ሰ(se) and ሠ(śe); ጸ(tṣ'e) and ፀ(ts'e) are Amharic alphabets currently having the same sound including their 7 consonant-vowel derivations. Even though there are standard rules for Amharic writing [3], online users tend to use homophone characters arbitrarily. The current trend in Amharic natural language processing (NLP) research is to normalize those homophone characters into a single representation [4]. This means, instead of the character ሀ(hä), ሃ(ha), ሐ(ḥä), ሓ(ḥa), ኀ(ḫä), ኃ(ḫa), and ኸ(ḫa), the character ሀ(hä) will be used; instead of አ(a), ኣ(ʔ), ዐ(ʔä), and ዓ(ʔa), the character አ(a) will be used; and so on. However, as far as we know, there are no studies to show that normalization is the right process to build semantic models for Amharic NLP tasks. Homophones with different symbols in Amharic literature might have different writing standards and different meanings from the language point of view [3]. So, when we apply normalization randomly, the homophone words will be changed into a single representation. And many words like መጽሐፍ (mets'əḥafə – book), ዓመት (ʔamatə – year), and ሥዕል (səʔələ – painting) will lose their standardized writing. The trend is narrowly approached to the normalization process as a "one-size-fits-all" task of replacing homophone variations with one representation [5].

To study the impact of homophone normalization, we have collected Amharic texts from different sources and built general-purpose pre-trained embedding models. We also analyzed the standard writing styles and studied the impacts of normalization on different semantic models. We have trained static word embedding models using word2Vec [6, 7] and fastText [8], and contextual embeddings using FLAIR [9] and RoBERTa [10] methods. Using these pre-trained embedding models, we have explored the impact of normalization on some of the Amharic NLP tasks, namely, part of speech (PoS) tagging, sentiment analysis (SA), and information retrieval (IR).

II. Amharic Language

Amharic (አማርኛ – amarəɲa) is written from left-to-right in Ge'ez alphabets called ፊደል (Fidäl) [11]. Ge'ez is a classical language of Ethiopia that belongs to the family of Semitic languages and has its alphabet (Fidäl) and numerical system. Fidäl is a syllable-based writing system where the consonants and vowels co-exist within each graphic symbol [12]. The syllabic change involves modifying the structure of the basic character by adding a small extension such as strokes, loops to the right, left, top, or bottom of the basic character, which helps to derive the other vowels [13].

[1] We have used the IPA notation for Amharic character transliteration

Amharic writing system adopted 26 basic characters from Ge'ez and added other 8 new characters ሸ(she), ኘ(nye), ቸ(che), ጨ(ch'e), ጀ(je), ቨ(ve), ዠ(zhe), and ኸ(ḫe). It has a total of 34 basic characters, where each letter (Fidäl) has 7 shapes or derivatives, with a total of 238 unique characters.

III. Overview of Homophone Normalization

Most previous approaches to Amharic normalization replace homophones with a single representation. For instance, one of the options was to replace the different homophones (e.g., ሀ(hä), ሃ(ha), ሐ(ḥä), ሓ(ḥa), ኀ(ḫä), ኃ(ḫa), and ኻ(ḫa)) with the first entry from the Amharic alphabet table[2] (e.g., ሀ(hä)).

Different scholars forwarded their proposals regarding Amharic homophone characters. The proposed arguments can be grouped into two categories. The first proposed argument is to represent homophones with a single orthography representation. This argument states that homophones are redundant alphabets, the homophone variations are problems for language models, and homophones in the orthography of Amharic cause issues in reading and teaching [4, 14, 15]. This argument is proposed based on the work of [16] regarding linguistic principles.

On the other hand, there is a strong recommendation for standardization in Amharic writing. Grammarians and historians proposed to adopt a strict usage of homophone characters to convey the correct meanings intended. The second argument states that Amharic should have consistent orthography and preserve all existing homophones according to its standard [3, 17, 18]. These scholars argued that standardization is better than normalization due to the following reasons. 1) Instead of avoiding the existing writing system, it is better to create a functional difference for each homophone. 2) Homophones do not cause much harm as they have different functions. 3) All previous documents are written using homophones and it is better to maintain the same standard for the new generations. 4) There are words written in different homophones that exhibit a difference in meanings.

Though there were some academic discussions on homophone characters, the suggested reform solutions were not largely successful in both proposed cases. Leaving this debate aside, in this work, we have assessed the status of homophones adoption in the online media text and the impacts of homophone character normalization on Amharic semantic models.

IV. Data Collection and Pre-processing

A. Amharic Corpus to Build Semantic Models

One of the main challenges for low-resource languages such as Amharic is the unavailability of a general-purpose corpus. The quality of the embedding models depends on the corpora size. We have collected a moderately large text collection from social media (Twitter around 2m sentences), news outlets (around 2m sentences), and web corpus[3,4] (2.5m sentences) with a total of 104m words from 6.5m sentences as shown in Table 1.

Table 1: Amharic corpus used to build embeddings

Sources	Sentences	Tokens
News	1,840,490	31,973,953
Twitter	2,131,879	35,109,854
Web corpus	2,463,471	39,326,483
Total	6,440,734	104,352,693

B. Pre-processing, Tokenization, and Segmentation

The development of every NLP component starts with data cleaning, language identification, tokenization, and segmentation in the pipeline. As the data is scraped from different sources, we have noticed several irregularities, where the texts include irrelevant content such as markup tags and special characters. There were also repeated sentences that are obtained from different sources. Thus, we performed series of preprocessing steps to canonize all tokens. During preprocessing, the following tasks have been performed: text cleaning, standardizing punctuation marks (for example convert all variants of full stops to a single one), removing non-Amharic texts, and word tokenization and sentence segmentation. This also includes removing URLs, HTML tags, scripts, and different boilerplate content. The Python "BeautifulSoup" library is used to clean most of the boilerplate content. The Python compact language detection (CLD2)[5] library is used to filter Amharic texts from the corpus.

Tokenization is one of the low-level NLP tasks, which is the split of text into recognizable tokens such as words and characters (punctuation marks), which is closely related to sentence segmentation. Text tokenization in Amharic is challenging, which cannot be achieved using the default "White Space Tokenizer" that is available in many frameworks such as NLTK and spaC[6]. Consider the following phrases that we have been retrieved from an online news portal:

- "አንቀጽ 28(3)(ሀ)፡" → [Article 28 (3) (a),]
- "ይናገር ደሴ (ዶ/ር) በ1996 ዓ.ም.‹‹1 ቁጥር፣ 2 ቁጥርና 3 ቁጥር የሠራተኛ ፍረጃ››ን" → [Yinager Dessie (Dr.) in 2004 ‹‹Number 1, Number 2, and Number 3 category of workers››"].

In the first example, a whitespace-based splitter results in 2 tokens while it has 9 tokens including the punctuation marks (አንቀጽ, 28, (, 3,), (, ሀ), and ፡). The second example is more complex, as we should consider abbreviations ዶ/ር (Dr.) and ዓ.ም (E.C), different quotation marks (‹‹ and "), years (1996), punctuation marks (፡), and named entities (ይናገር ደሴ – Yinager Dessie). For a proper tool to segment Amharic sentences and tokenize words, we have developed an Amharic segmenter and tokenizer using that can be integrated into the FLAIR framework. The Amharic tokenizer and segmenter tool is publicly available[7]

V. Semantic Models

One of the approaches of deep learning techniques is to use word embeddings to explore the semantic and syntactic relations of words. The embedding permits to capture of more

[2] The official name for the Amharic Alphabets table is 'Fidäl Gebeta'
[3] https://opus.nlpl.eu/
[4] https://github.com/adtsegaye/Amharic-English-Machine-Translation
[5] https://pypi.org/project/pycld2
[6] https://spacy.io/
[7] https://bit.ly/3zksf13

refined attributes and contextual cues that are inherent in human language. Semantics deals with the meaning of words, phrases, and sentences. In the context of this study, we define semantic models as the techniques and approaches used to build word representations that can be used in different downstream NLP applications. To develop the downstream NLP applications, we have built pre-trained embedding models from scratch using the collected corpus that we have discussed in Section IV–A. The semantic models include static word embeddings and contextual transformer-based embeddings.

A. Static Word Embeddings

Static word embeddings are classical representations, at the word level, where each distinct word gets exactly one pre-computed embedding representation. The only available pre-trained static word embedding for Amharic text is the fastText model, which is trained from Wikipedia and web data scraped in the common crawl project [19]. This embedding model is developed as part of multilingual setups, which will not fit the needs of most Amharic NLP tasks. For this study, we have computed word2Vec and fastText static word embeddings from scratch with the CBOW and Skip-gram approaches.

Word2Vec is a two-layer neural network that is used for training word representations with predictive language modeling. Its commonly used output is a vocabulary in which each item (word) has a vector attached to it, which can be fed into a deep-learning network or simply queried to detect relationships between words. We have built both CBOW and Skip-gram methods using 200 and 300-dimensional vectors. The CBOW model cares about the conditional probability of generating the central target word from given context words [7]. Skip-gram is the inverse of the CBOW that predicts the context from the target words [6]. We have built word2Vec embeddings using the Gensim Python Librar[8]. The parameters have been trained with a window of size 5, and a negative sample of size 10.

fastText is another word embedding method released by Facebook researchers [8], which is an extension of the word2Vec model that can construct embeddings for unseen words on the basis of their character n-grams. So, the vector for a word in fastText is made of the sum of character n-grams. We have developed both CBOW and Skip-gram techniques of fastText embeddings using 200 and 300-dimensional vectors. Our fastText embeddings are trained with parameters of window size of 5 and epochs of 10, as suggested by the original work [8].

B. Contextual Embeddings

RoBERTa: With the release of Google's Bidirectional Encoder Representations from Transformers (BERT) [20], Facebook researchers proposed an improved recipe for training the BERT model, RoBERTa [10], for the Robustly optimized BERT approach. RoBERTa is trained with dynamic masking patterns and full sentences without the next sentence prediction (NSP) of BERT. For this work, it has been trained using a GPU (NVIDIA Quadro RTX 6000 with 24GB RAM) using an "epoch" of 5 and a "block size" of 512.

FLAIR: It is a contextualized string embedding that is trained based on sequences of characters where words are contextualized by their surrounding characters and capture latent syntactic-semantic information [9]. Its API allows the application of the growing list of pre-trained embeddings for fine-tuning and includes methods for downloading standard NLP research datasets. We have trained FLAIR contextual string embeddings using a GPU server: the number of epochs is 50, mini-batch size of 32, and embedding size of 256.

VI. Related Works on Amharic NLP Tasks

In this section, we will discuss some of the related works for selected Amharic NLP tasks.

A. Part of Speech Tagging

Part-of-speech (PoS) tagging is the process of assigning grammatical categories to a morphological unit of a sentence. It is considered as one of the basic tools that are important for downstream NLP applications.

Several attempts have been made in the past to develop PoS tagger models for Amharic. The work by [21] attempted to develop a Hidden Markov Model (HMM) PoS tagger using 23 tags from 300 words. The work by [22] compared three tagging strategies, namely HMM, Support Vector Machines (SVM), and Maximum Entropy (ME) using the manually annotated corpus developed by [23] at the Ethiopian Language Research Center (ELRC) of Addis Ababa University. The dataset from ELRC contains 210,000 words from the news domain only. The work by [24] conducted PoS tagging experiments for Amharic using uncleaned ELRC corpus to use PoS information. Moreover, the work by [25] built a PoS tagging model, which results in a relatively good performance with small data set available for low-resource and morphologically rich languages. The work by [26] has attempted to extend the existing ELRC tag-set as well as increase the size of the corpus by incorporating Quran and Bible texts (ELRCQB). All of the above works did not explore the impacts of homophone normalization for Amharic PoS tagging.

B. Sentiment Analysis

Sentiment analysis is the task of detecting the orientation of someone's opinion and analyzing the emotions, feelings, and attitudes of a writer in a piece of information concerning a certain situation, object, or event [27]. It is a task of categorizing sentimental text in a specific document into 'positive' or 'negative' classes.

Some attempts have been made in the past to develop sentiment analysis for Amharic. The work by [28] describes a rule-based sentiment polarity classification system using movie reviews, where 955 sentiment lexicon entries are generated. The work by [29] presented a machine learning approach to multi-scale sentiment analysis on the Amharic language. This work tried to collect around 600 posts from online sources with only a limited diversity and using the Naïve Bayes algorithm. The work by [30] focused on the generation of the Amharic sentiment lexicon using the English sentiment lexicon. Lastly, the work by [31] explored sentiment analysis for the Amharic language based on the Twitter dataset using the FLAIR-based deep learning text classifier.

[8] https://radimrehurek.com/gensim/index.html

C. Information Retrieval

Information retrieval (IR) refers to the process, methods, and procedures of searching and retrieving recorded data or information from a file or database.

Some of the works done on Amharic IR's are the work by [32], where they developed a web search engine for Amharic web documents that has a crawler, an indexer, and a query engine component. The work by [33] designed an Amharic-English bilingual search engine based on the model that enables web users to find the information they need in Amharic and English languages. The work of [34] built the first reusable test collection for IR system benchmarking, but still, it is not publicly available. There is no prior work exploring if homophone normalization increases or decreases the performance of IR systems.

VII. Developing NLP Applications

After our pre-trained embeddings are built, we have used the open-source FLAIR framework [9] to fine-tune or customize our pre-trained embedding representations to build classification models for the selected downstream NLP tasks. We employ the current state-of-the-art approaches for sequence labeling NLP tasks, using the Bidirectional Long Short-Term Memory with a Conditional Random Field (BiLSTM-CRF) sequence labeling architecture. We used BiLSTM-CRF that was proposed by [35] and utilized the pre-trained embeddings [36] to address the sequence labeling tasks for PoS tagging. For the sentiment classification task, we have used the LSTM-based document classification algorithm from the FLAIR framework. We have experimented with both the regular and normalized embedding models and report the performance of each approach.

PoS tagging: For the PoS tagging experiment, we used the extended version of the ELRC dataset by [26] and we build a BiLSTM-CRF based PoS sequence tagger [35, 36]. The data is annotated with 66 tags. ELRCQB dataset from [27] has a total of 39k sentences (440,941 words) that are annotated for Amharic PoS tags using the 66 tags. We have split the PoS tagging dataset into training, testing, and development sets with the 80:10:10 splitting strategy.

Table 2 Experimental result for PoS classification

Models	Regular			Normalized		
	P	*R*	*F*	*P*	*R*	*F*
W2V_CBOW300D	84.2	70.1	**71.5**	82.0	69.2	70.3
W2V_Sg_300D	83.9	68.2	**69.5**	82.6	68.6	**69.5**
fT_CBOW300D	83.4	77.7	**75.7**	80.9	77.7	74.0
fT_Sg_300D	84.2	74.4	73.4	84.4	75.1	**75.2**
FLAIR	82.1	78.9	**77.5**	79.4	76.6	73.6

In Table 2: The models' names are abbreviated in the order of embeddings type, architecture type, and dimensions used. It shows the experimental results of the regular (without applying normalization) and normalized (with applying homophone normalization) embedding models.

Sentiment Analysis: For the sentiment analysis task, we have used the recently collected sentiment classification datasets, a total of 9.4k tweets where each tweet is labeled by 3 users [31]. The datasets are annotated in four sentiment classes, 'positive', 'negative', 'neutral', and 'mixed'. We have split the dataset into training, testing, and development sets with the 80:10:10 splitting strategy.

Table 3: Experimental result for sentiment classification

Models	Regular			Normalized		
	P	*R*	*F*	*P*	*R*	*F*
W2V_CBOW300D	57.6	46.4	**44.9**	61.0	44.5	43.9
W2V_Sg_300D	59.9	47.5	**49.1**	60.5	42.7	41.4
fT_CBOW300D	63.1	49.2	**50.7**	57.9	47.3	48.5
fT_Sg_300D	61.1	46.4	**46.9**	58.4	45.5	46.1
RoBERTa	61.8	55.5	57.3	61.40	58.7	**59.8**
FLAIR	57.0	56.5	**56.8**	47.0	56.9	56.6

Information Retrieval (IR): For the IR task, we have used the Elasticsearch document indexer[9]. Elasticsearch is a distributed, real-time, free, open-source search and analytics engine for all types of data, including textual, numerical, geospatial, structured, and unstructured [37]. We have indexed around 250,000 preprocessed regular sentences from our collected corpus. The sentences are indexed in regular and normalized fields. We have selected 10 queries to test the IR system as shown in Table 4. Regarding the queries selection, each query has been selected randomly by having at least one homophone character in each query. Searching is performed both in the normalized and regular indexes. When searching the normalized indexes, we have applied the same normalization strategy to the queries. Thus, the relevance of the results retrieved to the queries has been tested manually across all documents with the help of the Kiba[10] visualization tool.

Table 4: Search results of the 10 queries

Query	Regular indexed			Normalized indexed		
	P	*R*	*F*	*P*	*R*	*F*
1	94.3	92.4	93.4	94.7	97.2	95.9
2	70.8	78.7	74.5	72.4	86.5	78.8
3	47.9	79.0	59.7	63.6	84.5	72.6
4	76.5	85.2	80.6	78.7	90.0	83.9
5	17.6	82.2	28.9	95.7	98.4	97.0
6	88.6	93.5	90.9	88.9	94.8	91.7
7	98.2	81.2	88.8	96.7	97.8	97.2
8	82.9	89.9	86.2	83.3	92.4	87.6
9	91.3	96.1	93.6	91.3	96.0	93.6
10	96.7	97.3	96.9	96.9	97.9	97.4
Av.	76.5	87.6	81.6	**86.2**	**93.6**	**89.7**

Table 4 shows the precision, recall, and F1-score of the 10 selected queries[11] using keyword-based retrieval.

VIII. Discussion of Results

In this section, we will briefly discuss the results of the downstream NLP tasks regarding the impacts of normalization on the different semantic models. We have used a macro F1-score for the comparison of the models' performances.

[9] https://www.elastic.co/what-is/elasticsearch

[10] https://www.elastic.co/kibana

[11] **Queries**: 1-ጠቅላይ ሚኒስትር ዐቢይ አሕመድ, 2-በኮቪድ 19 የተያዙ ሰዎች, 3-መጭው አዲስ ዓመት, 4-የኢትዮጵያ ብሔራዊ ምርጫ, 5- የኢትዮጲያ ሕገ መንግሥት, 6-የብልጽግና ፓርቲ ማኒፌስቶ, 7-የሃይማኖት አባቶች, 8-የበሔር ብሔረሰቦች ቀን, 9-ተፎካካሪ የፖለቲካ ፓርቲዎች, 10-የሕዳሴው ግድብ ግንባታ

Part of Speech Tagging: As can be seen from Table 2, the PoS tagger based on the regular FLAIR and regular fastText (CBOW) models performs better, achieving an F1-score of 77.1% and 75.7% respectively, than the normalized taggers using the word2Vec and fastText embeddings. The normalized model based on the fastText (Skipgram) obtains a better result. When we analyze the test file, misclassified tags are not due to homophone character variations in the regular and normalized models, rather, it is due to the complexity of the Amharic PoS tagging task. For example, the word አንድያመጡት (ənədəyamet'utə – to bring it) has a gold label as "VREL – Relative verb", while it is predicted as "N – noun" using the regular models. Whereas it is predicted as "VP – Verb with a preposition" when the normalized model is considered. In contrast, we found out that words with different homophones are correctly predicted. For example, the standardized word በኀጢአታቸውም (beḫät'iatachewəmə – in their sin), has a gold label as "NPS – Proper noun plural" and predicted as "NPS – Proper noun plural" using the regular fastText (CBOW) model. Similarly, a normalized form of this word, በሀጢአታቸውም (behät'iatachewəmə – in their sin) is also predicted as "NPS – Proper noun plural" using the normalized fastText (CBOW) model. We also found out that the dataset has several inconsistencies and spelling mistakes. The quality of the dataset regarding annotation quality, spelling error, and many PoS tag classes (66 classes) will be another cause for the overall low performance of the PoS tagger. In general, we have observed that PoS taggers using the regular (unnormalized) embedding perform better than normalized models.

Sentiment Analysis: As shown from Table 3, the models based on regular texts have a better performance than normalized models.

Table 5: Error Analysis: normalized model misclassified

Tweets regular model correctly classifies	Anno.	Reg.	Norm.
አልሰማችሁም እምቢኝ አሉች ህወሓት (You have not heard, TPLF said no)	NEG	NEG	NEU
ነፃነትን የሚከለክል መንግስት ስለ ብልፅግና ቢያወራህ እንኳን ፅን ከመሀል እያወጣህ ስማው። ትክክለኛ ማንነቱ ያ ነው! (Even if a government that restricts freedom speaks to you about prosperity, listen carefully. That's the real identity!)	NEG	NEG	NEU
ምነው ቀልዱን ንፁሀን ላይ ባናረገው (Let's not put the joke on the innocent)	NEG	NEG	NEU

Table 5: shows the error analysis of tweet examples that are misclassified by normalized models. The "Anno" column is the annotated (labeled) class with Positive, Negative, Mixed, and Neutral. Reg. and Norm. columns are regular and normalized models respectively. NEG and NEU are negative and neutral classes respectively.

As we can see from Table 5, the three regular tweet texts contain the word ህወሓት (həweḫatə – TPLF), ብልፅግና (bələtsʼəgəna – prosperity), and ንፁሀን (nətsʼuhänə – innocent); and after applying normalization these three words are transformed into ህወሀት (həwehätə), ብልጽግና (bələtṣəgəna), and ንጹሀን (nətṣuhanə). In this situation, regular models classified these tweets correctly based on the annotated classes while the normalized models misclassified them.

Information Retrieval: For the IR system, we have evaluated the precision, recall, and F1-score of 10 keyword-based search queries. As we have seen from Table 4, more relevant documents have been retrieved from normalized indexed documents. When we use normalized queries to the normalized indexed document, we obtain more relevant documents. We have observed that normalization retrieves additional documents when the homophone words do not bear differences in meaning.

IX. Conclusion and Future Works

In this work, we have presented the first work on the impacts of homophone normalization on semantic models for Amharic. First, as more and more NLP applications have relied on deep learning approaches, we have collected and analyzed moderately large-scale text collection from Twitter, news portals, and web corpus. To study the normalization impact on NLP application models, we have build models using regular and normalized texts. The pre-trained embeddings that we have built include word2Vec, fastText, FLAIR, and RoBERTa embeddings, which are readily usable for any downstream NLP applications. Finally, we fine-tune the pre-trained embeddings and build part-of-speech tagging and sentiment analysis classification models using benchmark datasets. In both tasks, regular classification systems based on regular embeddings perform better than the normalized embeddings while normalization is essential for the information retrieval system.

We also explore the trends in online Amharic writing. Even though there are rules for Amharic writing, the online community tends to use homophone characters inconsistently. Our analysis from the collected text revealed that the online community does not follow the Amharic writing standard. We conclude that users are inconsistently choosing homophone characters due to the following reasons. 1) Lack of knowledge of the correct homophone for a given word. 2) Hindrance by technology (eg., computer or mobile keyboards might not support typing the different homophone characters). 3) Unable to understand the root word in the language (for example, most of the Amharic words have roots in Ge'ez Language). 4) Different backgrounds of the speaker (eg., Argobba, Harari, Siltie, Tigrigna). 5) tempted to use the most frequent character they are aware of.

The main contributions of this work are: 1) explore the impacts of normalization, 2) study the trends of Amharic writing style using online media text, 3) collection of Amharic text to train different embedding models, 4) Python-based preprocessing tools, 5) preparing benchmark dataset for Amharic PoS tagger and Sentiment analysis along with the classification models[12].

In the future, we will further explore the effects of homophone character normalization on more Amahric NLP tasks.

References

[1] A. M. Gezmu, B. E. Seyoum, and M. Gasser, and N. Andreas, "Contemporary Amharic corpus: Automatically morpho-syntactically tagged Amharic corpus," *In Proceedings of the First Workshop on Linguistic Resources for Natural Language Processing*, Santa Fe, NM, USA, pp. 65-70, 2018.

[12] All resources: https://bit.ly/3zksf13

[2] A. Salawu and A. Aseres, "Language policy, ideologies, power and the Ethiopian media," *Communication 41(1),* vol. 41, no. 1, pp. 71-89, 2015.

[3] R. Cowley, "The standardisation of Amharic spelling," *Journal of Ethiopian Studies,* vol. 5, no. 2, pp. 1-8, 1967.

[4] S. T. Abate, M. Woldeyohannis, M. Y. Tachbelie, M. Meshesha, S. Atinafu, W. Mulugeta, Y. Assabie, H. Abera, B. Ephrem, T. Abebe, W. Tsegaye, A. Lemma, T. Andargie, and S. Shifaw, "Parallel corpora for Bi-Lingual English-Ethiopian languages statistical machine translation," *In Proceedings of the 27th International Conference on Computational Linguistics*, Santa Fe, NM, USA, 2018, pp. 3102-3111.

[5] B. Tyler and L. Yunyao, "An In-depth Analysis of the Effect of Text Normalization in Social Media," *In Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Denver, Colorado, 2015, pp. 420–429.

[6] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *In Proceedings of the 1st International Conference on Learning Representations*, ICLR13, Scottsdale, AZ, USA, arXiv:1301.3781, 2013.

[7] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2, NIPS'13, Lake Tahoe, NV, USA,* pp. 3111–3119, 2013.

[8] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information, " *Transactions of the Association for Computational Linguistics,* vol. 5, pp. 135-146, 2017.

[9] A. Akbik, T. Bergmann, D. Blythe, K. Rasul, S. Schweter, and R. Vollgraf, "FLAIR: An easy-to-use framework for state-of-the-art NLP," *In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, Minneapolis, MN, USA, 2019, pp. 54-59.

[10] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "RoBERTa: A robustly optimized BERT Pretraining Approach," *arXiv:1907.11692,* 2019.

[11] A. Eshetu, G. Teshome, and T. Abebe, "Learning Word and Sub-word Vectors for Amharic (Less Resourced Language)," *International Journal of Advanced Engineering Research and Science (IJAERS),* vol. 7, no. 8, pp. 358-366, 2020.

[12] S. T. Abate and W. Menzel, "Syllable-based speech recognition for Amharic," *In Proceedings of the 2007 Workshop on Computational Approaches to Semitic Languages: Common Issues and Resources*, Prague, Czech Republic, 2007, pp. 33-40.

[13] F. Negesse and D. Ado, "Visual recognition of graphic variants of Amharic letters: Psycholinguistic experiments," *Oslo Studies in Language,* vol. 8, no. 1, pp. 173-199, 2016.

[14] F. Menuta, "Over-differentiation in Amharic orthography and attitude towards reform," *The Ethiopian Journal of Social Sciences Language Studies,* vol. 3, no. 1, pp. 3-32, 2016.

[15] E. Tadesse, R. Tsegaye, and K. Qaqqabaa, "Event Extraction from Unstructured Amharic Text," *In Proceedings of The 12th Language Resources and Evaluation Conference*, LREC20, Marseille, France, pp. 2103-2109, 2020.

[16] J. Berry, "The making of alphabets," *Readings in the Sociology of Language*, De Gruyter Mouton, pp. 737-753, 2012.

[17] A. M. Gezmu, B. E. Seyoum, T. T. Lema, and A. Nürnberger, "Manually annotated spelling error corpus for Amharic," Technical report (Internet), Data and Knowledge Engineering Group, Fakultät für Informatik, Otto-von-Guericke-Universität Magdeburg, 2017.

[18] D. Yacob, "Application of the double metaphone algorithm to Amharic orthography," arXiv:cs/0408052.

[19] E. Grave, P. Bojanowski, P. Gupta, A. Joulin, and T. Mikolov, "Learning word vectors for 157 languages," *In Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, 2018, pp. 7-12.

[20] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Minneapolis, MN, USA., 2019, pp. 4171–4186.

[21] M. Getachew, "Automatic part-of-speech tagging for Amharic language an experiment using stochastic Hidden Markov Approach," Master's thesis, Information Sciences, School of Graduate Studies, Addis Ababa University, 2001.

[22] B. Gambäck, F. Olsson, A. A. Argaw, and L. Asker, "Methods for Amharic part-of-speech tagging," *In Proceedings of the First Workshop on Language Technologies for African Languages*, Athens, Greece, 2009, pp. 104–111.

[23] G. A. Demeke and M. Getachew, "Manual annotation of Amharic news items with part-of-speech tags and its challenges," In proceedings of *Ethiopian Languages Research Center Working Papers*, vol. 2, Addis Ababa, Ethiopia, 2006, pp. 1-16.

[24] M. Y. Tachbelie and W. Menzel, "Amharic part-of-speech tagger for factored language modeling," *In Proceedings of the International Conference RANLP-2009*, Borovets, Bulgaria, 2009, pp. 428-433.

[25] M. Y. Tachbelie, S. T. Abate, and L. Besacier, "Part-of-speech tagging for underresourced and morphologically rich languages—the case of Amharic," *In Proceedings of conference on Human Language Technology for Development*, *HLTD (2011), Alexandria, Egypt ,* pp. 50-55, 2011.

[26] I. Gashaw and H. L. Shashirekha, "Machine Learning Approaches for Amharic Parts-of-speech Tagging," arXiv:2001.03324, 2020.

[27] P. Pandey and S. Govilkar, "A framework for sentiment analysis in Hindi using HSWN," *International Journal of Computer Applications,* vol. 119, no. 19, pp. 23-26, 2015.

[28] B. G. Gebre, "Part of speech tagging for Amharic," PhD dissertation, School of Law, Social Sciences and Communications, University of Wolverhampton, Wolverhampton, UK, 2010.

[29] W. Philemon and W. Mulugeta, "A Machine Learning Approach to Multi-Scale Sentiment Analysis of Amharic Online Posts," *HiLCoE Journal of Computer Science Technology,* vol. 2, no. 2, pp. 1-8, 2014.

[30] G. N. Alemneh, A. Rauber, and S. Atnafu, "Dictionary Based Amharic Sentiment Lexicon Generation," *In Proceedings of International Conference on Information and Communication Technology for Development for Africa*, ICT4DA 2019, Bahir Dar, Ethiopia, 2019, pp. 311-326.

[31] S. M. Yimam, H. M. Alemayehu, A. A. Ayele, and C. Biemann, "Exploring Amharic Sentiment Analysis from Social Media Texts: Building Annotation Tools and Classification Models," *In Proceedings of the 28th International Conference on Computational Linguistics*, Barcelona, Spain, 2020, pp. 1048-1060.

[32] T. Mindaye and S. Atnafu, "Design and implementation of Amharic search engine," *In Proceedings of 2009 Fifth International Conference on Signal Image Technology and Internet Based Systems*, Marrakech, Morocco, 2009, pp. 318-325.

[33] M. Munye and S. Atnafu, "Amharic-English bilingual web search engine," *In Proceedings of International Conference on Management of Emergent Digital EcoSystems*, *MEDES '12,* Addis Ababa, Ethiopia, 2012, pp. 32-39.

[34] T. Yeshambel, J. Mothe, and Y. Assabie, "2AIRTC: The Amharic Adhoc Information Retrieval Test Collection," *In Proceedings of International Conference of the Cross-Language Evaluation Forum for European Languages*, Switzerland, 2020, pp. 55-66.

[35] Z. Huang, W. Xu, and K. Yu. Bidirectional LSTM-CRF models for sequence tagging, arXiv:1508.01991, 2015.

[36] A. Akbik, D. Blythe, and R. Vollgraf, "Contextual string embeddings for sequence labeling," *In Proceedings of the 27th international conference on computational linguistics*, Santa Fe, NM, USA, 2018, pp. 1638-1649.

[37] C. Gormley and Z. Tong, *Elasticsearch: the definitive guide: a distributed real-time search and analytics engine*. " O'Reilly Media, Inc.", 2015.