Corpus Generation to Develop Amharic Morphological Segmenter

Terefe Feyisa, Dr Seble Hailu
Information Network Security Administration, Addis Ababa, Ethiopia

***Abstract*—Morphological segmenter is an important component in Amharic natural language processing systems. Despite this fact, Amharic lacks large amount of morphologically segmented corpus. Large amount of corpus is often a requirement to develop neural network-based language technologies. This paper presents an alternative method to generate large amount of morph-segmented corpus for Amharic language. First, a relatively small (138,400 words) morphologically annotated Amharic seed-corpus is manually prepared. The annotation enables to identify prefixes, stem, and suffixes of a given word. Second, a supervised approach is used to create a conditional random field-based seed-model (on the seed-corpus). Applying the seed-model (an unsupervised technique on a large unsegmented raw Amharic words) for prediction, a large corpus size (3,777,283) of segmented words are automatically generated. Third, the newly generated corpus is used to train an Amharic morphological segmenter (based on a supervised neural sequence-to-sequence (seq2seq) approach using character embeddings). Using the seq2seq method, an F-score of 98.65% was measured. Results show an agreement with previous efforts for Arabic language. The work presented here has profound implications for future studies of Ethiopian language technologies and may one day help solve the problem of the digital-divide between resource-rich and under-resourced languages.**

***Keywords*—Amharic; Amharic morphology; segmentation corpus; seq2seq; under-resourced languages**

I. Introduction

Language plays a significant role in achieving the sustainable development goals (SDG 2030) [1]. Regarding language technologies, the Prime Minister of Ethiopia, Dr Abiy Ahmed, recently said, "(. . .) teaching Somali, Tigrinya, Amharic and Oromo languages with artificial intelligence and making these languages researchable is a great achievement (. . .)" [2]. This Prime Minister's quote can be interpreted as artificial intelligence (AI) in general, and natural language processing (NLP) in particular, are issues of contemporary importance in Ethiopia. This work focuses on one aspect of Amharic NLP technology.

NLP aims at enabling machines to understand human languages. Machines usually obtain "natural language" in the form of voice or text messages. Typically, textual and vocal data are not structured and therefore require advanced technologies, like deep neural networks (DNNs), to be used and understood correctly. DNNs are mainly based on large amount of corpus for automatic feature extraction [3]. Because of this (large corpus requirement), DNNs are not being fully applied by almost all under-resourced Ethiopian languages.

Despite being the working language of Ethiopia, Amharic is one of the under-resourced languages [4], [5]. Being under-resourced, Amharic lacks digital resources, such as sizable segmentation corpus and a morphological analyzer [6].

The lack of digital resources is mainly attributed to an expensive corpus preparation by language experts. Depending on their level of expertise, a linguist may ask starting from Ethiopian Birr 10.00 per a single word segmentation. For example, in 2019 there was a joint project (between the former Information Network Security Agency and Addis Ababa University). The aim of the project was to develop core NLP tools. The biggest share of the project cost was the payment for the linguists. Even with the minimum price, Birr 10.00 per a single word segmentation, the cost gets in millions just for about 100,000 distinct word segmentation.

The problem of language corpus scarcity is a "real-life" challenge that exists when developing NLP tools and undertaking researches.

One of the primary consequences of corpus scarcity is reflected in the approaches to develop Amharic NLP technologies. The dominant approaches to develop Amharic NLP systems are mostly rule-based, such as memory-based learning [7].

Rule-based systems have their own pros and cons [8]. Rule-based systems are advantageous, as they are declarative and are easy to comprehend, to maintain, to incorporate domain knowledge, and to trace and fix the cause of errors. However, they are heuristic and require tedious manual labor as compared to machine-learning (ML) approaches.

ML-based systems too have their own pros and cons [8]. On their advantage end, they are: trainable, adaptable, and reduces manual effort. Their disadvantages includes the requirement of: labeled corpus, retraining for domain adaptation, ML expertise to use or maintain, and they are opaque.

Given the situations, it is essential to design an alternative mechanism to enrich (with corpus), and develop language processing tools for Amharic (to make it researchable). One mechanism could be the design of an algorithm that is computationally robust and less expensive.

To design such an algorithm, a possible approach would be the use of a hybrid system: a combination of rule-based and ML-based systems. The rule-based system can be applied to generate seed-corpus for a semi-supervised ML approach as suggested by [9]: "Minimally supervised approaches provide better performance compared to applying only unsupervised methods on large unlabeled datasets."

The purpose of this work is twofold. First, to automatically generate morph-segmented Amharic corpus. Then, the newly

generated corpus is used to develop a neural network-based Amharic morphological segmenter (AMS).

To the best of our knowledge, there has not been any work that attempted neural network techniques to develop AMS by using semi-supervised learning approach to automatically generate large amount of corpus.

The main contributions are the following:

1) An alternative algorithm is used to construct a morphologically segmented corpus for Amharic. The corpus is annotated with boundaries that clearly mark prefix, stem, and suffix morphemes.
2) A sequence-to-sequence neural network approach is used to create an Amharic morphological segmenter.
3) The research shall motivate the understanding of (the processing challenges of) Amharic.
4) The research shall inspire further research (by releasing the resources – the corpus and the algorithm – of this research to the public).

The organization of this paper is as follows: Section II provides an overview of recent advancements in morphological segmentation. Section III outlines our methodology. Subsequently, Section IV presents the results obtained from testing the method on diverse subjects. Finally, the conclusion summarizes the findings of this study and offers insights into future perspectives.

II. Related Work

A. Why Develop Amharic Corpora

Amharic corpora is useful to develop, and apply a research work for different Amharic natural language technologies. In 2005, [6] manually annotated Amharic words (in news documents) with the most appropriate parts-of-speech (POS) tags. They managed to annotate 1,065 text documents having 202,671 words [10]. Their corpora is useful to develop probabilistic POS tagger [11] and chunker [10].

In 2016, a semi-automatic approach (very similar to this work) is followed by [12] to develop a morpho-syntactically annotated Amharic Treebank to develop a text parser. They first annotated 1,000 sentences for POS tags, morphological information, and syntactic relations of words. Using these sentences as seed-corpus, they trained a machine learning system to automatically annotate 5,000 sentences.

In 2021, [13] developed a POS tagged corpus consisting of 25,199 documents using syntactic information of words. Their corpus was tagged automatically using HornMorpho analyzer [14] with manual intervention to correct erroneous results. The morphological analyzer generates the derived stems of non-verbal words rather than basic stems. For verbs, it generates only roots rather than stems producing incorrect representations. Their corpus is not directly suitable for morphological segmentation experiments. Nevertheless, one can use their corpus as part of a seed-corpus by appropriating to a desired experiments. Regardless, HornMorpho is used by most works related to Amharic morphological segmentation [15], [16]. It is a fully-fledged morphological analysis tool for Amharic, Tigrinya and Oromo languages.

The work of [17] is also worth mentioning as, they used morphological knowledge and an extension of existing annotated dataset to improve the performance of an Amharic POS tagging system.

This paper's approach is different from HornMorpho. In that it is limited to only word segmentation task (as opposed to HornMorpho, a fully-fledged morphological analysis tool).

Brief, although morphological properties have been used to create POS tagged corpora, there is no morphologically annotated large Amharic corpus to date (i.e. that can be directly consumed by a neural network model). This study aims to construct a hybrid system to generate a morphologically annotated segmented words that can be useful for sequence-to-sequence neural network models.

B. Segmentation for Semitic Languages

Word segmentation is regarded as a first step for almost all Semitic languages [18], [19]. This work adopts most of the methods presented for Arabic word segmenter [20]. Their method involves three steps. First, they used a small manually segmented Arabic corpus (110,000 words) to create a "seed-model". Then, they used the "seed-model" to bootstrap an unsupervised algorithm. Finally, they applied the unsupervised algorithm on a large unsegmented Arabic corpus (155 million words). They claimed a 97% exact match accuracy on a test corpus (28,449 words). A significant difference between this work and theirs is the choice of an unsupervised algorithm. Theirs is a "trigram language model". This study used a conditional random field (CRF) instead. The CRF model is a relatively better algorithm (e.g., it can use language-independent features of characters, in addition to n-grams) [18].

C. Sequence-to-Sequence Approaches

Recently, supervised sequence-to-sequence approaches have gained success [21]–[23]. The seq2seq modeling of this work is mostly inspired (and adapts most of the techniques used) by a morphological segmentation task for the Russian language [24]. The Russian work defined MS as sequence transduction using character embeddings. They used the architecture and the hyperparameters by [22].

D. Summary

This work builds on previous works on morphological segmentation, such as by [20]. It then enriches an already existing, manually segmented seed-corpus by applying a tool – mostly used for Amharic NLP works [14]. Finally, it adapts a sec2seq model by [24].

III. Methodology

Amharic morphological segmentation can be modeled using only hand-crafted dataset [8]. However, building a sizable hand-crafted corpus is expensive in the amount of human work. AMS can also be designed automatically using statistical models such as Hidden Markov Model (HMM) and Conditional Random Fields (CRF) [25]. Today, one can also apply deep learning methods to get a state-of-the-art performance level [26].

This work attempts to combine the best of rule-based, ML, and deep learning approaches. To that end, it follows a three-step process (see Fig. 1).

First, a supervised approach is applied to create a **seed-model** using a hand-crafted dataset (training a CRF). Then, based on the seed-model a CRF-based unsupervised method is applied on a **raw unsegmented words** to **enrich** the manually created Amharic corpus. Third, a supervised neural **sequence-to-sequence** (seq2seq) learning approach, using character embeddings, is implemented for AMS on the enriched dataset. The seq2seq is mostly inspired by the work of [20] and the soft-attention encoder-decoder research method of [27].

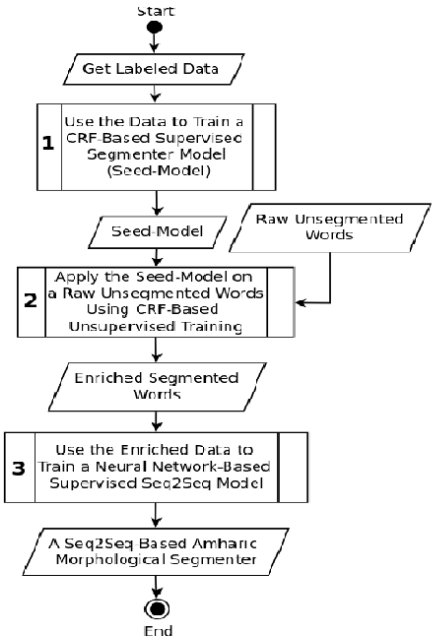

Fig. 1. A flowchart to highlight the three major sub-processes of the proposed method.

A. Seed-Corpus for the Seed-Model

Two kinds of dataset are used to prepare a seed-model. The first one is a manually labeled, morphologically segmented corpus (173,000 words), prepared by [28].

The 173,000 segmented corpus is used for three purposes. First, it helped in generating an affixation table. Second, 80% of it (138,400 words) is used as a part of the training corpus. Third, 20% of it (34,600 words) is used as test corpus for the unsupervised CRF-based model.

The corpus by [28], however, has only representative *stems*. To compensate for the lack of *stem varieties*, another dataset, from Contemporary Amharic Corpus (CACO) by [4] is used.

Basically, the CACO corpus is a morphological analysis result of HornMorpho [14]. As such, it is not directly applicable for the purpose of this study. So, it is filtered by applying a regular expression algorithm and the affixation table to get 906,417 words.

Finally, the two dataset (manually segmented corpus (138,400 words) and the filtered corpus (906,417 words)) are merged to get a total of 1,044,817 words (see Table I) as a seed-corpus to train a seed-model. All the 1,044,817 words are labeled using the "BMES" tagging scheme.

TABLE I. SUMMARY OF SEED-CORPUS PREPARATION

CORPUS	WORDS
MANUALLY SEGMENTED	138,400
FILTERED FROM CACO	906,417
MERGED TOTAL (SEED-CORPUS)	**1,044,817**

B. The BMES Tagging Scheme

Training a word segmenter can be considered as an organized classification task with encoded classes [18].

An encoding is used to identify the presence of morph boundaries around a target character. Models using fine-grained tagging schemes contribute significantly for performance accuracy [29], [30]. As such, this work adopts the fine-grained "BMES" encoding scheme by [31]. This encoding scheme uses four class set {B, M, E, S} to capture information about the sequence of morphs in a given word. The labeling symbols have the following meanings:

(B)egin: The start of a morph.
(M)iddle: The continuity of a morph.
(E)nd: The end of a morph.
(S)ingle: Single morphs.

Table II depicts an instance of the "BMES" tagging scheme using an example of three Amharic words (/bet/ "*house*", /betu/ "*the house*", and /betunmko/ "*and also the house*") with their corresponding manual segmentation (marked by a dash "-") and labeling.

TABLE II. AN INSTANCE USAGE OF THE "BMES" TAG SCHEME

SEGMENTED WORD	LABELING
bet	[B, M, E]
bet-u	[B, M, E, S]
bet-u-n-m-ko	[B, M, E, S, S, S, B, E]

C. Training a Seed-Model

The seed-corpus (1,044,817 words, labeled with the "BMES" tagging scheme) is used as an input for a supervised CRF training to prepare a seed-model. The linear-chain CRF model of the Wapiti toolkit [32] is used for both segmenting and labeling purposes. The CRF model takes a labeled seed-corpus, a template to mark n-grams and a text file to write the output (the seed-model). The seed-model's accuracy is tested to be 96% exact match accuracy on a manually segmented test corpus of 34,600 words.

D. Corpus Generation from a Bulk Corpus

New corpus generation, from a bulk corpus, demands the seed-corpus, bulk-corpus and an affixation table. Having those prerequisites, a four step process follows. First, the seed-corpus is generated. Then, a bulk-corpus is prepared from the *most frequently used Amharic word lists* (347,039 **unsegmented** words) [33]. Third, a seed-model is **trained** using the seed-corpus. Finally, using the seed-model (iteratively) **re-training** is performed by using a block from the bulk-corpus. Fig. 2 is a flowchart to illustrate the process. (see also, Algorithm 1 on the following page, for detailed steps.)

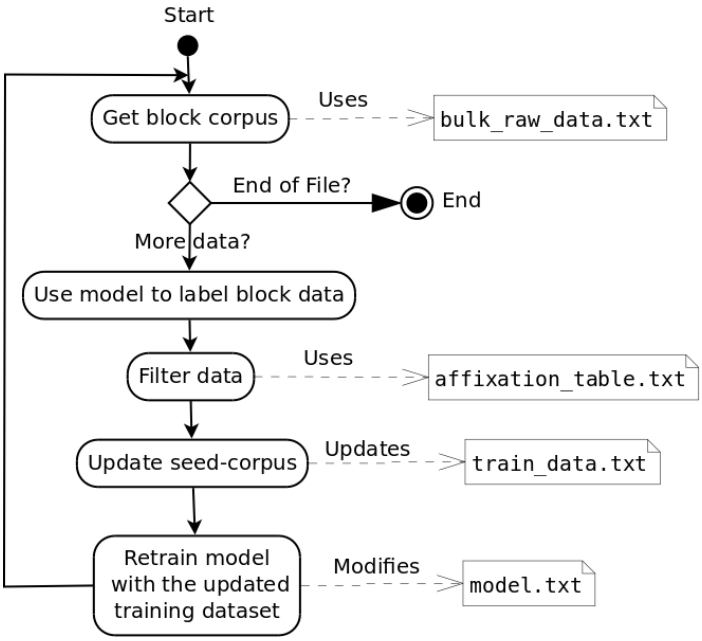

Fig. 2. Corpus preparation from a raw unsegmented corpus.

Using the algorithm, 2,732,466 segmented words are filtered from a bulk raw dataset. This corpus size (2,732,466 words) together with the seed-corpus (1,044,817) help in training the seq2seq AMS model (see Table III).

TABLE III. SUMMARY OF THE NEW CORPUS SIZE USING THE ALGORITHM

CORPUS	WORDS
SEED-CORPUS	1,044,817
CORPUS FROM BULK-CORPUS	2,732,466
MERGED DATASET FOR SEQ2SEQ	**3,777,283**

E. The Seq2Seq Model

The seq2seq works as transduction system [24]. That means, AMS as a seq2seq model gains an overall information from inputs and directly output a segmented sequence without using context features.

Table IV presents a sample input-output pair for a seq2seq AMS model. The input, $\boldsymbol{X} = x_1, x_2, x_3, x_4$, is the word "betu", "the house" having 4 characters ($x_1 = b, x_2 = e, x_3 = t, x_4 = u$). The output is a sequence, $\boldsymbol{y}$, having 5 characters including a boundary marker, β.

The final segmentation result is {bet}β{u}.

TABLE IV. INPUT-OUTPUT INSTANCE FOR SEQ2SEQ AMS MODEL

	Sequence	Length
Input	$\boldsymbol{X}$ = [b, e, t, u]	4
Output	$\boldsymbol{y}$ = [b, e, t, β, u]	5

The attention-based seq2seq model architecture for AMS is shown in Fig. 3. The model contains character embedding layer, an encoder layer, and an attention head and a decoder.

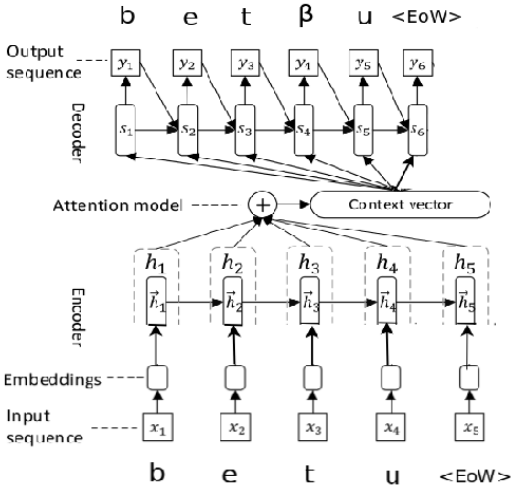

Fig. 3. Architecture of the seq2seq model (Adapted from [21]). <EoW> stands for End-of-Word.

F. Settings

Dataset: The total corpus size is 3,777,283 morphologically segmented Amharic words. This corpus is divided into two parts: 90% for training, and 10% for testing as suggested by [24]. The targets for the seq2seq model are morph-broken words (see Table V).

Algorithm 1 Get New Corpus From Bulk Corpus

Input: seedCorpus, bulkCorpus, affixationTable

```
blockSize ← 1000
begin ← 0
end ← blockSize
seedModel ← crfTrain(seedCorpus, 'model.txt')
newCorpus ← seedCorpus
while begin ≤ sizeOf(bulkCorpus)
do  wordBlock ← getCorpusBlock(bulkCoupus, begin, end)
    for each word ∈ wordBlock
    do  transliteratedWord ← transliterate(word)
        writeOnFile(transliteratedWord, 'transliteratedWord.txt')
        crfPredict('model.txt', 'transliteratedWord.txt', 'result.txt')
        filteredSegments ← filterSegments('result.txt', affixationTable)
        newCorpus ← merge(newCorpus, filteredSegments)
        newCorpus ← dropDuplicates(newCorpus)
    newModel ← crfRetrain(newCorpus, 'model.txt')
    begin ← end+1
    end ← end + blockSize
return (newCorpus)
```

TABLE V. SAMPLE DATASET FOR SEQ2SEQ TRAINING AS A PAIR OF [INPUT WORD, TARGET SEGMENTED WORD]. THE EXAMPLE AMHARIC WORD HAS A ROOT **SBR**, HAVING THE SENSE OF BREAKING

INPUT WORD	SEGMENTED WORD
s b a b r o n a	s b a b r - o - n a
s b a b r a n a	s b a b r - a - n a
s b a b r w n a	s b a b r - - w - n a

Packages The Python programming language is applied for the experimentation. As for the deep learning package, Keras [34] with TensorFlow [35] as a back-end is used. For evaluating the models, the Scikit-learn [36] toolkit is used.

G. Train the Seq2Seq Model

The seq2seq model involves three distinct models: an encoder, an attention head and a decoder. Each of these (three) models, include a single Bidirectional Long Short Term Memory (BiLSTM) layer.

- The Long Short Term Memory (LSTM) Network
  The LSTM network architecture consists of a set of recurrently connected memory blocks, known as LSTM memory cells (dotted boxes in Fig. 4).
  LSTM are better at finding and exploiting long range dependencies in a data [37]. It has an input layer **X**, hidden layer **h** and output layer **y**. For example, if one inputs $x = [b, e, t, u]$, *the house*, into the LSTM, the expected prediction is a tag set as: $y = [B, M, E, S]$ (see Fig. 4).

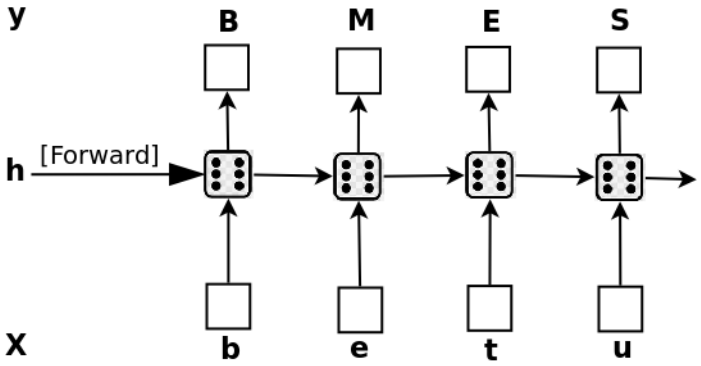

Fig. 4. The LSTM network.

- The Bidirectional LSTM
  As depicted in Fig. 5, BiLSTM is two hidden LSTM layers. In sequence tagging task, it enables us to have access to both past and future input features.

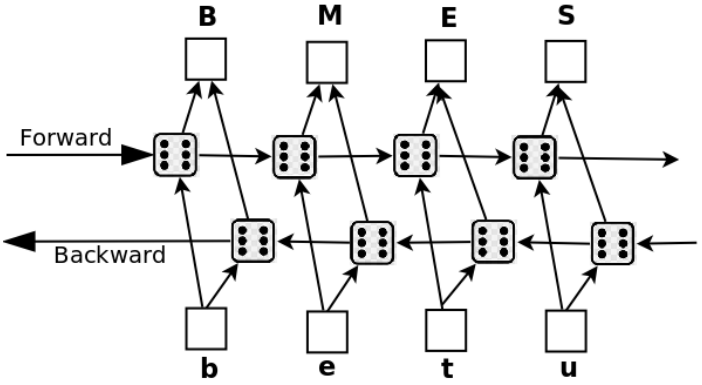

Fig. 5. A BiLSTM network.

- Hyperparameters
  - Inspired by previous works [22], [38], possible parameter combinations are explored in the preliminary experiments. The complete list of parameters is shown in Table VI. "Hidden layer size" stands for the number of BiLSTM layers or hidden state dimension and "Embeddings dimension" stands for dimensionality of the embedding layer.

TABLE VI. HYPERPARAMETERS OF THE SEQ2SEQ TRAINING.

HYPERPARAMETER	VALUE
Encoder and decoder	
Number of epochs	10
Number of units	1024
Batch size	64
Character embeddings	256
Optimizer	Adam
Attention	
Attention type	Bahdanau's

- Training
  - The training involves three distinct models (an encoder, an attention head and a decoder, in that order) acting as a single end-to-end model.

    Encoder
    It takes a list of tokens. It converts those tokens into vectors by an embedding layer. Then, a BiLSTM layer processes the vectors sequentially. It outputs the processed sequence (for the attention head) and the internal state (useful to initialize the decoder).
    Bahdanau's additive attention [27]
    It computes the attention weights and the context vectors.
    Decoder
    After accepting the output from the encoder, it converts the tokens into a vector using an embedding layer. The decoder keeps track of what has been generated so far using a similar layer as in the encoder. Finally, it produces context vectors and do "logit" predictions for the next token.

H. Evaluation

Two morphological segmentation evaluation approaches are suggested by [39]. The first one is called "direct evaluation", in which the results of a MS model are compared to "gold" standards. The other approach is known as "indirect evaluation", where the MS models are used in other applications such as for speech recognition system.

As recommended by [24], this study uses the direct evaluation technique. So, boundary precision, boundary recall and boundary F1-score are reported.

$$\text{Precision} = \frac{\text{number of correct boundaries found}}{\text{total number of boundaries found}} \quad (1)$$

$$\text{Recall} = \frac{\text{number of correct boundaries found}}{\text{total number of correct boundaries}} \quad (2)$$

$$\text{F1-score} = \frac{2\times \text{ Precision}\times \text{ Recall}}{\text{Precision + Recall}} \quad (3)$$

Where, "boundary" means the border between morphs. For example, suppose there are two boundaries in the gold standard for the Amharic word "y-bet-u" (of-the-house). If the AMS model segments this word as "y-be-t-u", with three boundaries, one can compute precision as 67%, recall as 100% and F1-score as 80%.

IV. EXPERIMENTAL RESULTS

A. Results

Overall, 3,777,283 morphologically segmented Amharic words are generated with an algorithm that involves a CRF model. The CRF model's accuracy was 96% exact match on a manually segmented test corpus of 34,600 words.

This accuracy is slightly less than that of the Arabic word segmenter by [20], which was 97% exact match accuracy on a test corpus (28,449 words). The difference may be attributed to the use of a large unsegmented Arabic corpus (155 million words) as compared to 347,039 unsegmented Amharic corpus.

Once the morphologically segmented Amharic words are generated, the next step was to develop our seq2seq model. But, before developing the seq2seq model, the newly generated data is used for training LSTM, GRU, BiLSTM, and BiGRU models in order to choose the best performing one. The performance of the models was evaluated and contrasted (see Table VII). The results showed that BiLSTM gave the best performance compared to the other models. So, we have chosen the BiLSTM model to implement our seq2seq model.

TABLE VII. A COMPARISON: TO CHOOSE THE BEST PERFORMING MODEL

MODEL	PRECISION	RECALL	F1-SCORE
GRU	91.73%	92.56%	92.14%
LSTM	92.47%	93.36%	92.91%
BiGRU	95.58%	95.95%	95.76%
BiLSTM	**98.47%**	**98.84%**	**98.65%**

Fig. 6 presents the attention weights for the input characters "slvbetacnnmko", where the sound /v/, representing //, is used for technical reason. The output is the "morph broken characters" which is found to satisfy the given "gold" standard "/slv-bet-acn-n-m-ko/".

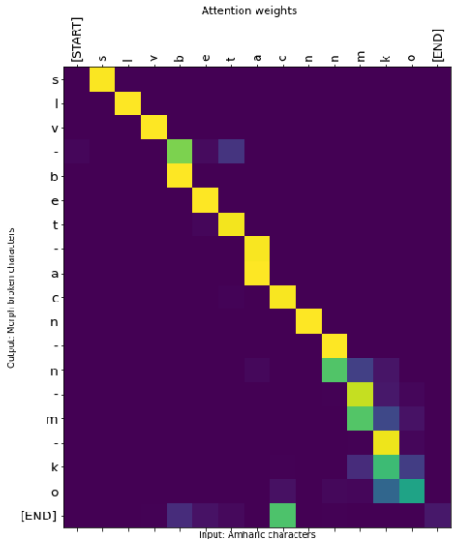

Fig. 6. Attention weights with inputs (Amharic characters) and outputs (morph broken characters) of the inputs.

B. Discussion

For accurate 'interpretation' of the human language, machines should be equipped with effective natural language processing components [40]. But, this is not often the case for under-resourced languages.

Under-resourced languages are the majority of the world languages, which have not attracted much attention from researchers and donors due to economical and political reasons [40]. For these languages, corpus is a challenge to train deep learning models [41], as deep learning techniques demand large amounts of labeled corpus [3].

Amharic is one of the under resourced Ethiopian languages [4] that lacks necessary corpus and NLP applications.

To improve the situation, an Amharic morphological segmenter is implemented based on a supervised neural sequence-to-sequence approach using character embeddings, by carefully constructing language resources. But, there are still possible error sources that put our results questionable.

One possible source of error is the use of a segmentation corpus from an external morphological analyzer (HornMorpho). Errors may propagate from the morphological analyzer to the filtered corpus. However, it is difficult to spot out the exact source of errors, as this work lacks a proper error analysis.

Nevertheless, the obtained F1-Score (98.65%) indicates that, there is a window of opportunity to improve the accuracy of the Amharic morphological segmenter by applying deep neural networks.

The main focus of this work was on corpus preparation. As such, this work lacks testing and comparison of the implemented supervised neural sequence-to-sequence (seq2seq) system against the stated previous works and with the resource-rich languages.

Comparing the obtained results with another similar implementation would have a much more impact to outline the achieved milestone. So, this can be considered as yet another limitation of this work.

One of the strongest suits of this research is the use of different datasets for seed-corpus preparation (as, having datasets for under-resourced languages is the main obstacle when it comes to the implementation of morphological segmentation systems).

V. Conclusion

Unlike morphologically poor languages, such as English, Amharic language's word segmentation resources aren't sufficient for researchers to do their practices.

Addressing the most understudied corpus generation for a low-resource language is fascinating, and is a big step for further studies.

Using annotated datasets and unsupervised techniques, a relatively big dataset was generated. This enabled the implementation of a seq2seq-based morphological segmenter for Amharic language.

Rule-based approach is used alongside a supervised machine learning approach. This hybrid system is found to be cost-effective, flexible, and most importantly effective in constructing a language resource for segmentation.

To construct a language resource, three sources of dataset are used. The first set is 138,400 manually labeled, morphologically segmented corpus. The second source is a morphological analysis result from Contemporary Amharic Corpus (filtered using a regular expression algorithm and a rule-based method by using an affixation table) to get 906,417 words. The third source is the result of applying a corpus generator algorithm out of "*most frequently used Amharic word lists*". Using the third technique, 2,732,466 segmented words are generated.

The newly generated segmentation corpus is then used to train a morphological segmenter model based on a supervised seq2seq neural network approach.

The seq2seq implementation involves three models appearing as one: an encoder, an attention head, and a decoder. For the seq2seq implementation, Python programming language is used on an Ubuntu machine having 64GB of memory.

The implemented seq2seq model is evaluated using a direct evaluation technique. Besides the 3,777,283 Amharic language corpus generated in the process, a 98.47% precision, a 98.84% recall, and a 98.65% F1-score have been achieved.

Brief, the major findings of this work are:

- An alternative algorithm that uses small seed-corpus to generate a large dataset from a raw bulk corpus.
- The generation of 3,777,283 morphologically segmented Amharic word corpus.
- An implementation of a seq2seq-based Amharic morphological segmenter model using the newly segmented word corpus.

- A state-of-the-art accuracy of the seq2seq morphological segmentation model (F1-score of 98.65%).

ACKNOWLEDGMENT

Our deep gratitude goes to Dr Derib Ado and Dr Demeke Asres Ayele who offered us valuable corpus and links. Our heartfelt appreciation goes to Dr Yemane Keleta Tedla, who sent us his full PhD dissertation paper on Tigrinya morphological segmentation.

REFERENCES

[1] D. Traoré, "The role of language and culture in sustainable development," 11 2017.

[2] N. Tube, "Dr abiy ahmed speech on artificial intelligent," 2020.

[3] H. Liang, X. Sun, Y. Sun, and Y. Gao, "Text feature extraction based on deep learning: a review," *Eurasip Journal on Wireless Communications and Networking*, vol. 2017, 2017.

[4] A. Mekonnen, M. Gasser, A. Nürnberger, and B. Seyoum, "Contemporary amharic corpus: Automatically morpho-syntactically tagged amharic corpus," 10 2018.

[5] P. Rychlý and V. Suchomel, "Annotated amharic corpora," in *Text, Speech, and Dialogue* (P. Sojka, A. Horák, I. Kopeček, and K. Pala, eds.), (Cham), pp. 295–302, Springer International Publishing, 2016.

[6] T. Yeshambel, J. Mothe, and Y. Assabie, "Morphologically annotated amharic text corpora," in *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, (New York, NY, USA), p. 23492355, Association for Computing Machinery, 2021.

[7] M. Abate and Y. Assabie, "Development of amharic morphological analyzer using memory-based learning," in *Proceedings of the 9th International Conference on Natural Language Processing (PolTAL2014)*, vol. 8686, pp. 1–13, Springer Lecture Notes in Artificial Intelligence (LNAI), 2014.

[8] L. Chiticariu, Y. Li, and F. R. Reiss, "Rule-based information extraction is dead! long live rule-based information extraction systems!," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, (Seattle, Washington, USA), pp. 827–832, Association for Computational Linguistics, Oct. 2013.

[9] T. Ruokolainen, O. Kohonen, K. Sirts, S.-A. Grönroos, M. Kurimo, and S. Virpioja, "A comparative study on minimally-supervised morphological segmentation," 2015.

[10] G. Demeke and M. Getachew, "Manual annotation of amharic news items with part-of-speech tags and its challenges," 01 2006.

[11] M. Tachbelie, S. Abate, and L. Besacier, "Part-of-speech tagging for under-resourced and morphologically rich languagesthe case of amharic," 04 2011.

[12] B. Seyoum, E. Binyam, Y. Miyao, B. Mekonnen, and Yimam, "Morpho-syntactically annotated amharic treebank," 06 2016.

[13] A. M. Gezmu, B. E. Seyoum, M. Gasser, and A. Nürnberger, "Contemporary amharic corpus: Automatically morpho-syntactically tagged amharic corpus," *CoRR*, vol. abs/2106.07241, 2021.

[14] M. Gasser, "Hornmorpho 2.5 users guide," 2012.

[15] A. T. Gebru and Y. Assabie, "Development of amharic grammar checker using morphological features of words and n-gram based probabilistic methods," in *Proceedings of the The 13th International Conference on Parsing Technologies (IWPT2013)*, pp. 106–112, 2013.

[16] T. Dawit and Y. Assabie, "Amharic anaphora resolution using knowledge-poor approach," in *Proceedings of the 9th International Conference on Natural Language Processing (PolTAL2014)*, vol. 8686, pp. 278–289, Springer Lecture Notes in Artificial Intelligence (LNAI), 2014.

[17] I. Gashaw and H. L. Shashirekha, "Machine learning approaches for amharic parts-of-speech tagging," *CoRR*, vol. abs/2001.03324, 2020.

[18] Y. Tedla and K. Yamamoto, "Morphological segmentation with lstm neural networks for tigrinya," *International Journal on Natural Language Computing*, vol. 7, pp. 29–44, 04 2018.

[19] M. Walther, "Computational nonlinear morphology with emphasis on semitic languages," *Computational Linguistics*, vol. 28, pp. 576–581, 12 2002. George Anton Kiraz (Beth Mardutho: The Syriac Institute) Cambridge: Cambridge University Press (Studies in natural language processing, edited by Branimir Boguraev and Steven Bird).

[20] Y.-S. Lee, K. Papineni, S. Roukos, O. Emam, and H. Hassan, "Language model based arabic word segmentation.," pp. 399–406, 2003.

[21] X. Shi, H. Huang, P. Jian, Y. Guo, X. Wei, and Y.-K. Tang, "Neural chinese word segmentation as sequence to sequence translation," in *Communications in Computer and Information Science*, pp. 91–103, Springer Singapore, 2017.

[22] D. Britz, A. Goldie, M.-T. Luong, and Q. Le, "Massive exploration of neural machine translation architectures," *arXiv preprint arXiv:1703.03906*, 2017.

[23] T. Ruzsics and T. Samardžić, "Neural sequence-to-sequence learning of internal word structure," in *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, (Vancouver, Canada), pp. 184–194, Association for Computational Linguistics, Aug. 2017.

[24] N. V. AREFYEV, T. Y. GRATSIANOVA, and K. P. POPOV, "Morphological segmentation with sequence to sequence neural network," in *Komp'juternaja Lingvistika i Intellektual'nye Tehnologii*, pp. 85–95, 2018.

[25] N. Ljubešić, "Comparing crf and lstm performance on the task of morphosyntactic tagging of non-standard varieties of south slavic languages," in *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)*, pp. 156–163, 2018.

[26] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *Journal of Machine Learning Research*, vol. 12, no. 76, pp. 2493–2537, 2011.

[27] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," 2014.

[28] D. Ado, "Amharic morph order and concatenation rule." unpublished, 2021.

[29] Y. Kitagawa and M. Komachi, "Long short-term memory for japanese word segmentation," 09 2017.

[30] J. Yang, S. Liang, and Y. Zhang, "Design challenges and misconceptions in neural sequence labeling," *CoRR*, vol. abs/1806.04470, 2018.

[31] T. Ruokolainen, O. Kohonen, S. Virpioja, and M. Kurimo, "Supervised morphological segmentation in a low-resource learning setting using conditional random fields," in *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, (Sofia, Bulgaria), pp. 29–37, Association for Computational Linguistics, Aug. 2013.

[32] T. Lavergne, O. Cappé, and F. Yvon, "Practical very large scale CRFs," in *Proceedings the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 504–513, Association for Computational Linguistics, July 2010.

[33] P. Rychlý and V. Suchomel, "Annotated amharic corpora," vol. 9924, pp. 295–302, 09 2016.

[34] Google, "Keras: The python deep learning library," 2020.

[35] Google, "Tensorflow: An end-to-end open source machine learning platform," 2020.

[36] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[37] J. Brownlee, "Time series prediction with lstm recurrent neural networks in python with keras," 2020.

[38] E. Ansari, Z. Žabokrtský, M. Mahmoudi, H. Haghdoost, and J. Vidra, "Supervised morphological segmentation using rich annotated lexicon," in *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, (Varna, Bulgaria), pp. 52–61, INCOMA Ltd., Sept. 2019.

[39] S. Virpioja, V. Turunen, S. Spiegler, O. Kohonen, and M. Kurimo, "Empirical comparison of evaluation methods for unsupervised learning of morphology," *Traitement Automatique des Langues*, vol. 52, pp. 45–90, 01 2011.

[40] J. Muhirwe, "Towards human language technologies for under-resourced languages," 2007.

[41] Y. Roh, G. Heo, and S. E. Whang, "A survey on data collection for machine learning: a big data-ai integration perspective," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 4, pp. 1328–1347, 2019.