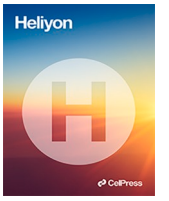
Heliyon 9 (2023) e17175

Contents lists available at ScienceDirect

Heliyon

journal homepage: www.cell.com/heliyon



Improving part-of-speech tagging in Amharic language using deep neural network

Sintayehu Hirpassa [a,*], G.S. Lehal [b]

[a] Department of Computer Science, Adama Science and Technology University, Ethiopia
[b] Department of Computer Science, Punjabi University, India

ARTICLE INFO

Keywords:
Natural language processing
Conditional random fields
Recurrent neural network
Long short-term memory
Feature representation

ABSTRACT

To date, several POS taggers have been introduced to facilitate the success of semantic analysis for different languages. However, the task of POS tagging becomes a bit intricate in morphologically complex languages, like Amharic. In this paper, we evaluated different models such as bidirectional long short term memory, convolutional neural network in combination with bidirectional long short term memory, and conditional random field for Amharic POS tagging. Various features, both language-dependent and -independent, have been explored in a conditional random field model. Besides, word-level and character-level features are analyzed in deep neural network models. A convolutional neural network is utilized for encoding features at the word and character level. Each model's performance has evaluated on the dataset that contained 321 K tokens and manually tagged with 31 POS tags. Lastly, the best performance obtained by an end-to-end deep neural network model, convolutional neural network in combination with bidirectional long term short memory and conditional random field, is 97.23% accuracy. This is the highest accuracy for Amharic POS tagging task and is competent with contemporary taggers currently existing in different languages.

1. Introduction

Part-of-Speech (POS) tagging is the task of classifying words in a sentence into their lexical categories (or parts of speech), based on their identical linguistic behaviors or syntactic context of the word [1]. POS tagging is the primary task in speech recognition, named entity recognition, shallow parsing, etc. [2,3]. The performance of POS tagging is thus exceptionally imperative for such applications since it enormously affects the effectiveness of the steps in the pipeline further. Analyzing POS tag of each token in a text is often a nontrivial task as the token might be ambiguous, which means some of the tokens could be belonging to more than one class due to the context they have [4]. Moreover, some speeches may be complex or unspoken [1]. The arrangement and organization of a language, such as Amharic, are associated with a set of specific rules and protocols that determine how words are grouped into phrases, how phrases are combined into clauses, and how clauses are merged into sentences. These regulations and practices are important notions in the creation of a POS tagger.

POS tagging has significantly improved in several languages as a result of the emergence of deep learning. To be specific, recent studies in different languages [5,6] show that deep neural network (DNN) architectures become an effective tool for a sequential tagging problem. Generally speaking, three benefits make the deep learning technique a preferable choice for sequential labelling

* Corresponding author.
E-mail addresses: sintsha2002@gmail.com (S. Hirpassa), gslehal@gmail.com (G.S. Lehal).

https://doi.org/10.1016/j.heliyon.2023.e17175
Received 20 March 2021; Received in revised form 8 June 2023; Accepted 9 June 2023
Available online 21 June 2023

tasks. First, the POS tagging problem benefits from the non-linear transformation as it maps from the input to the output [7]. DNN framework can learn intricate data features through nonlinear activation functions contrary to linear models like HMM/CRF [7]. Second, it saves considerable effort to develop features for POS tagging problems. On the contrary, the classical ML approaches require extensive engineering skills and field expertise [7,8]. Third, through gradient descent, the DNN allows an end-to-end paradigm in POS tagging system development. These possessions make it possible to design an advanced POS tagging system for complex and NLP resource-scarce languages.

The existing POS tagger trained by DNN model on the WSJ dataset has achieved human-level accuracy of 98.85% [9]. On the contrary, achieving human-level accuracy in a morphologically complex and highly inflected language like Amharic remains an unsolved problem. This work is based on the recently published paper by Mequanint [10], where deep learning was successfully used for Amharic POS tagging by exploiting the available labelled corpora of Amharic language. The author implemented a BiLSTM based POS tagging and obtained 93.67% performance on an Amharic dataset. As a limitation of Mequanint's [10] work: they used merely word embeddings. They didn't take character-level features that could be valuable for rare words and make the feature set rich. Yet, to make one tagger the most effective, morphological/orthographical (character-level) features must take into consideration and select which features are more important for the task at hand [11]. Thus, in this study, we have investigated a CNN in seeking to tackle the issue of character-level feature representation. We were inspired by Refs. [11,26], in these works, a character-embedding was applied for POS tagging in English language and they achieved better performance.

The main contributions of this work are fourfold:

- ~ Extending the existing dataset into 321 K part of speech labelled words
- ~ Comparing the performances of three neural network models: BiLSTM, BiLSTM + CNN, BiLSTM + CNN + CRF with a classical machine learning algorithm, CRF for the Amharic POS tagging.
- ~ Investigating a character-level feature and showing CNN method is efficient in character-level feature learning.
- ~ Proposing an architecture that can easily fit other sequential labelling tasks, like NER in Amharic language.

1.1. Motivation of study

Three core problems can be seen as constraints in POS tagging in Amharic. Spelling variations or lack of spelling standardization. The same word could appear in different spellings with the same sound. For example, the word "sun" can be written in various ways in Amharic (ፀሐይ/ጸሀይ/ጸሐይ/ፀሀይ), which makes the task difficult to distinguish each word by the POS tagger. However, this problem can be solved by using character-level embedding. The other one is the unavailability of standard language resources. As mentioned above, Amharic language suffers from a lack of NLP resources. For instance, there is no benchmark Amharic POS tagged dataset for machine learning as other language datasets integrated with NLTK, UCI, Kaggle, 20 Newsgroups, toward data science etc. Furthermore, the sentence/phrase formation structure in Amharic can be inconsistent, different words have different meanings in different positions. This situation has a profound effect during POS tagging. Thus, we can tackle such issues by analysing deep-level (both semantic and syntactic) features.

Although, recently few POS tagging experiment has been conducted for Amharic, exerting continuous effort is necessary to develop a tagger that can achieve the same level of accuracy and reliability as those used in English and other languages. This work thus focused on developing a POS tagging strategy that would have phenomenal performance. There are seven works (POS tagging in Amharic) that have been done so far, however, none of which has become available for public use. This is due to two reasons: First, they are incomplete regarding the dataset they used. Second, performance issues must be paramount as POS tagger is the backbone of most NLP application development. In addition, except for one work, all works have analyzed the classical machine learning models for this task (please refer to the succeeding section for a detailed description). We believe that this is the major reason why the works hadn't achieved great accuracy. Therefore, in this study, we are motivated to evaluate the deep learning method since it is the current state-of-the-art learning approach for sequence labelling problems.

1.2. Amharic language and related works

Amharic is the main language widely used in Ethiopia and is a member of the Semitic branch of the Afro-Asian superfamily [12]. Particularly, two major factors could affect the development of any NLP tool for Amharic language. The first one is the absence of sufficient NLP related resources and tools. But, recently, Amharic has made some major steps forward, with a medium-sized POS tagged corpus [13] and a morphology analyser available for the public [14]. The second issue is that the language has a syntactic context sensitivity: the orthography of the letter relies on the letter's position in the text. The Amharic language is notably rich in morphology and inflection, with a significant amount of non-vocabulary words [15]. Despite these hindrances, some investigators have tried their unremitted efforts for developing different NLP applications for this language.

The efforts of POS tagging in Amharic context date back to 2001 reported by Getachew [17]. The model was implemented using HMM, and 25 POS tags were compiled for the first time. These tags provided the foundation for other researchers. Here, a one-page long corpus was used for the experiment as a training set; and reported 87% F-measure. While considering the size of the training data they used, difficult to conclude that the performance obtained by the model was rational. As a result, several investigators, both native and non-native speakers, have expressed great enthusiasm and have proceeded to build a POS tagger using a range of classical machine learning models [16–22].

Adafre [18] examined the application of CRF for POS tagging in Amharic for the first time. He composed five News texts containing 1000 entries and tagged them manually with only 10 POS tags. Manually crafted features, including character features, morphological features, dictionary features, the previous tag, and character binary features were considered for the tagger's development. Finally, the proposed model based on CRF achieved 74% F1 score. This result showed that the performance could get to an adenoidal degree if the linguistic resources are comprehensive. He also explicitly claimed that large-sized Amharic tagged data is necessary to meet good performance with state-of-the-art taggers. Based on this and other findings, Getachew & Demeke [13] then took the move on Amharic tagged corpus development. The researchers gathered news articles in Amharic language, comprising 1065 sentences and 210 K tokens, from Walta Information Center (WIC), a domestic news department in Ethiopia. These were subsequently labelled with 30 POS tags through manual means.

Later, Gamback et al. [16] initiated to make a comparison of three ML models for POS tagging in Amharic. These were HMM-based Trigram'n'Tags (TnT), SVM-based SVMTool, and Maximum Entropy-based MALLET. These models were evaluated on a corpus that was already prepared by Getachew and Demeke [13]. They used a set of manually crafted features including morphological, orthographic, and syntactic information. In this, the researchers also mapped their tags by decreasing the previous tags from 30 to 11 main classes to examine how much the tagger's accuracy is influenced by the number of POS tags. Each model gave a good accuracy, although the SVM-based model had the highest performance among them, obtaining 88% F1 score on 30 tags and 92.77% F1 score on 11 tags. According to the researcher, although a model based on TnT has less accuracy, it is more efficient in memory requirement and processing time.

In the same year, Tachbelie et al. [20] also worked on POS tagging for Amharic texts. They employed the same corpus as previous work and attempted to explore the effectiveness of two well-known taggers such as SVMTool and TnT. Based on their results, they reported that the SVM-based tagger outperformed the HMM-based tagger with 85.5% and 82.99% F1 scores, respectively. If we recall the previous work of Gambäck [16], SVM had similarly outperformed the models it was compared to.

Afterwards, Gebre [19] proposed a successful tagger, where he used ML models. A range of experiments have been conducted using Brill, CRF, SVM, and TnT models, along with handcrafted features, and evaluated the proposed taggers on the ELRC [13] corpus. In this work, the researcher made a lot of improvements in the existing corpus, even though we were not able to get it for our experiment. Similar to previous machine learning based studies, various features were used and novel features were also added, like vowel patterns information in a word and radicals. Finally, the CRF-based model outperformed the others, yielding 90.95% F1 score. The author demonstrated that the amount of the training dataset, the tagset, and the feature selected are key factors that significantly impact the model's efficiency and effectiveness. Various studies also support the fact that the tagging task does not rely solely on the dataset used in the training process, but a feature set and tagset are equally important.

Tachbelie et al. [21] explored four tagging strategies including CRF, SVM, MBT, and TnT to investigate the best tagger for Amharic. They derived the main classes of POS tags among 31 POS tags and evaluated each model by varying the training set at each stage. In addition, they investigated the impact of segmenting the preposition and conjunction in words on the tagger's accuracy. Finally, they reported that MBT is a good tagger for Amharic, as its performance is less affected while the amount of training data grows compared with other taggers, especially with TnT. Besides, word segmentation has shown promising ways to augment the performance of POS tagging in Amharic. Going further, recently, Mequanint [10] came up with a novel strategy for POS tagging in Amharic. He tried to eliminate the manual feature generation process to develop tagging models; implementing an automatic way of generating features called word embedding. This feature was a way to easily capture the syntactic and semantic information of words. The experiments were conducted using a DNN architecture called BiLSTM and achieved a maximum accuracy of 93.67% on the ELRC corpus. Moreover, he revealed that the segmentation of words had improved the overall accuracy by 5%.

The remaining section of this article is structured as follows. Section 2 Experimental methodology**.** Section 3 describes Deep Learning for POS tagging. Section 4 describes a detailed description of the proposed DNN model. The details of the experiments and results are discussed in Section 5. Section 6 Discussion based on CRF and CNN-BiLSTM-CRF model and analysis of POS tagging errors. Towards the end, Section 7 conclusion and future directions of this study are discussed.

2. Experimental methodology

2.1. Conditional random fields

CRFs are a statistical modelling approach for labelling sequential data. Lafferty et al. [22] introduced CRFs in 2001. Since then, CRF is applied widely in the areas of computer vision, NER, parsing, and particularly for sequence labelling problems [23]. Structurally, it can be characterized as an undirected graphical model, constructed with nodes, which contain the sequence labels *Y* along with observing sequence *X*. The Conditional Distribution can be modelled by Equation (1) as follows:

$$\hat{y} = argmax_y P(y / x) \tag{1}$$

In CRF, the input data is sequence-based and essentially the previous inputs must be considered when making predictions about the current data points. To model this, the functions for feature $f(X, i, l_{i-1}, l_i)$ will be used, where X the set of input vectors, the position *i* of the data point, l label data, the purpose of the feature function is to express some kind of characteristic of the sequence that the data point represents. For instance, for POS tagging, then: f (X, i, l{i 1}, l{i}) = 1 if l {i 1} is a Noun, and l {i} is a Verb. 0 otherwise. Similarly, f (X, i, l {i 1}, l {i}) = 1 if l {i 1} is a Verb and l {i} is an Adverb. 0 otherwise [22]. While building the conditional field, each feature

function is assigned a set of weights that the algorithm will analyze it. The Probability distribution for CRFs can be represented mathematically in Equations (2) and (3) as follows:

$$P(y,X,\lambda)=\frac{1}{Z(X)}\exp\left\{\sum_{i=1}^{n}\sum_{j}\lambda_j f_i(X,i,y_{i-1},y_i)\right\} \tag{2}$$

where:

$$Z(x)=\sum_{y'\varepsilon y}\sum_{i=1}^{n}\sum_{j}\lambda_j f_i\left(X,i,y'_{i-1},y'_i\right) \tag{3}$$

2.1.1. Features for CRF learning

Apparently, feature-based probabilistic models demand multiple features which enable training rapidly and minimize the intricacy of a model. Consequently, feature selection is a crucial aspect since it influences the tagging accuracy of the model. Therefore, the choice of features should be meticulous in CRFs too. In this experiment, various candidate features have been analyzed to prove the most important features for POS tagging development using CRF model. The features listed for CRF based model development are as follows,

Contextual features: Like any other language, Amharic is also suffering from the word ambiguity problem. For example, see these two sentences:

(A) "የጀርባው በር"/The back door;
(B) "ከኔ ጀርባ ላይ/On my back.

In these sentences, the word "ጀርባ/back" may receive two sorts of speeches; it takes an adjective in sentence A and a noun in sentence B. To resolve such kind of ambiguities, we must define context words with some window size. Similarly, POS information of the current and context words is considered.

Morphological features: As mentioned above, Amharic is a morphologically rich language. Thus, discovering the prefix and suffixes of a word is a means to minimize arbitrary tagging, as this feature is beneficial to identify the POS classes. In a language that is highly inflected and agglutinative, like Amharic, the morphological features have been found effective in POS tagging tasks, as Molina [33] stated. Based on this fact, we have applied this feature for POS tagging in Amharic.

- *Orthographic features:* It is a binary feature that has two possible outcomes and it records data that explains the shape that is present in the word. A few of the orthographic features are:
- *Comprised numerals:* This is a feature that has a value of either true or false and is employed to verify if the present symbol is composed solely of numerals. It assists in identifying numerical words, which are mainly utilized for numbering tags.
- *Comprises special symbol:* This binary value function is built in to check if the current token contains special characters (%, $, etc.), which are used for recognizing symbol tags.
- *Word Position:* This can be either a real-valued or a binary feature, determining the index of each word in a sentence, for example, "beginning of the sentence (BOS)" or "end of the sentence (EOS)".

3. Deep learning for POS tagging

Akin to the CRF model, DNN formulates the POS tagging task into a sequence labelling problem. As Young et al. [25] review that in several sequential labelling tasks like NER and POS tagging, a straightforward DNN could achieve human-level accuracy. Very recently, multiple complex DNN based POS taggers have become influential and render forward-looking outcomes in morphological rich language too. As a result, in this study, we employ DNN models, i.e., multiple variants of RNN such as LSTM and BiLSTM, and CNN for a character-level feature for POS tagging in Amharic.

3.1. Convolutional neural network (CNN)

In this network, a variety of distinct layers is widely used. These are convolutional layer, pooling layer, ReLU layer, and fully connected layer. The first layer is the convolution, the centrepiece of a CNN formation. This layer contains a number of learnable filters, having a small receptive field which may be extended over the full depth of inputs. The second layer is pooling. It is a nonlinear down-sampling. This layer is mainly essential to reduce the representation's spatial size gradually, the number of parameters, and the computational time. There are three nonlinear functions for applying to the pool. These are mean, average, and maximum. The third layer is the rectified linear unit (ReLU), which is an activation function that effectively confiscates negative values from an activation map by converting them to zero. The final one is the fully connected layer. The high-level perception in the network will be performed via completely connected layers, after many convolutional and max-pooling layers.

3.2. Long short term memory (LSTM)

LSTM is a promising recurrent architecture, which is capable of bridging long delays among significant input and output occurrences, thus accessing long-range context events has become trivial [30,31]. LSTM is particularly designed to solve some recognized issues related to exploding and vanishing gradients by affixing an extra memory cell [30]. Long-distance dependencies are quite captured by LSTMs. It receives a sequence of vectors (×1,×2, …,xn) of length n as input and produces a hidden state output sequence of vectors (h1,h2, …,hn). Each block comprises one or more self-connected memory cells and three multiplicative units, such as the input, output and forget gates [31], that provide constant analogues of storage and access information over long periods of time. Due to this, data dependencies are possibly defined and exploited better, as LSTM is the most prominent RNN architecture for sequence labelling problems. Comprehending the advantages, we apply this architecture to the Amharic POS tagging problem.

3.3. Bidirectional long short-term memory (BiLSTM)

As we mentioned above, the stream of information in LSTM is forward through a looped structure, while in BiLSTM, the network can capture the properties of the sequenced data in forward, backward direction to gain and embrace the context-based evidence for long periods [28]. Thus, BiLSTM removes the issue of partial context that smears to feed-forward model [3].

3.4. Feature representation for deep neural network

3.4.1. Word and character-level embedding

Word embedding is implemented by following the distributional theory: "The terms with similar meanings usually occur in similar contexts". This similarity is determined by measuring the vector similarity using either cosine similarity or other techniques. Word embedding does not require a costly annotated corpus, it can rather be easily extracted from vast, readily available unannotated corpora [34]. Word embedding is recently proven to be efficient in computing core NLP tasks, perhaps among the crucial breakthroughs for the imposing accomplishment of DNN models on arduous NLP application development [35]. The limitation of word embedding is that it ignores or disregards intra-word information like morphology and orthography of the word. Notably, in the POS-tagging task, the morphological and orthographic features remain valuable [26]. Therefore, we should look for learning at the character-level representation of the words.

We used CNN for character-level representation. Various studies, such as [26,27,32] have shown that CNN is a practical approach to extract the morphological information from characters of words and encode it unto neural representations. This representation produces a set of tag distributions for each character, and then, a word-level tag will be mapped from the tags. Recently, character level feature representation is taking a big deal of attention for POS tagging tasks [9,11,26].

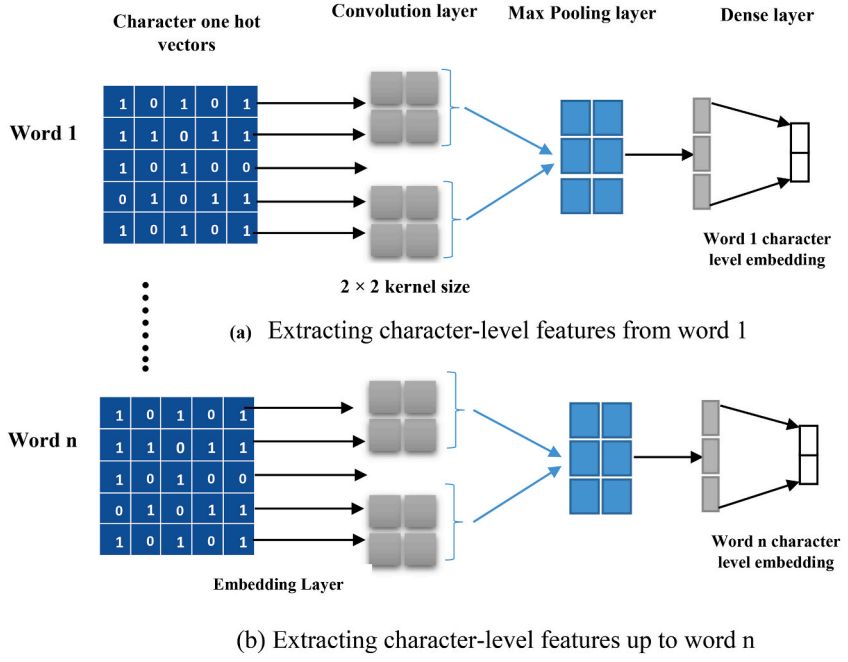


Fig. 1. Visualizes how CNN converts each text into one hot vector and extract character-level feature from each input word (x_i) in a sentence using various layers. Different colors designate different-sized layers. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

4. Proposed model: a deep neural network

At this point, we put forth our proposed model inspired by Chiu & Nichol's work [27] for NER application. To attain optimal outcomes, we combine BiLSTM and CRF models with embeddings curated at word and character levels in a network, as depicted in Fig. 2. We used a bidirectional LSTM network to get word sequences for outstanding foresight. The proposed model is a complete end-to-end solution that does not necessitate any task-specific feature engineering and/or data pre-processing. The CNN-BiLSTM-CRF architecture, as illustrated in Fig. 1 comprises three tiers. The first tier is responsible for accepting the inputs simultaneously. At the outset, every term in a given sentence $\boldsymbol{x}$ is described as an embedding, which is a real-valued feature vector that also includes the word's character embedding.

To maintain the semantic and syntactic details of words, every word in the input series is characterized by a fusion of two feature vectors: character-level and word-level embeddings. We initialize word-level embedding to precisely capture the semantic meaning of words. To grasp orthographic characteristics, we employed a CNN framework as used by Ma and Hovy [32] and Santos et al. [26]. As depicted in Fig. 1, every term is symbolized by a matrix of $\boldsymbol{v} \times \boldsymbol{l}$ dimensions, where $\boldsymbol{v}$ denotes the character's vocabulary size. Then, it mixes into a matrix of $\boldsymbol{c} \times \boldsymbol{d}$ dimensions, where $\boldsymbol{c}$ stands for the character counts and $\boldsymbol{d}$ is the embedding dimension. Four convolutional filters are utilized to seize character n-grams of varying dimensions. Subsequently, the outputs of every convolutional filter are directed to a max-pooling layer, and the pooling outputs are merged to represent the term. We use four different filter widths in the convolution process to capture character n-grams of different window size features. A concatenated n-gram character embedding is employed to represent the context semantics. To be more precise, adjacent character embedding will be merged with, specifically, unigrams and bigrams. We used a stacked CNN with n-gram filters (e.g., n = 2, 3, 4) for character-level feature extraction. We also employed a fixed-size word vocabulary [***Vwrd***] and a fixed-size character vocabulary [***Vchr***], which is created by extracting them from the corpus provided for this study. The character level embedding vector is combined with the word embedding vector in the form of [**rwrd, rwch**]. This vector then becomes an input to a BiLSTM. The outputs of both the forward and backward sequences, $_{h}$ and $_{h'}$, will be merged through an activation function and provided to a CRF layer.

The activation function ReLU exists at each hidden layer, which is used to calculate the weights for each input. The motivation for choosing this activation is, empirically, previous studies witnessed that training a deep network with ReLu tends to converge more rapidly (more computationally efficient) and reliably than sigmoid and tanh [41]. Since Relu only needs to choose max $(0, x)$, and not carry out costly exponential operations as in sigmoid, where it does not activate all neurons simultaneously. This means that the neurons will be deactivated if the outcome of the linear transformation is less than zero. Due to this reason, throughout the back-propagation process, the biases and weights of some neurons are not updated.

CRF Layer: at the output layer of the network, Instead of making an independent label prediction, CRF is implemented to decode the accurate labels among all possible label sequences. Thus, the values predicted by the activation function at the BiLSTM layer will be provided as input to CRF layer for further processing. Unlike an LSTM network, BiLSTM-CRF doesn't produce label probabilities for single words; instead, it calculates the scores of the whole set of labels in a particular sentence concurrently. The score function for an LSTM and linear chain CRF can be described by Equation (4).

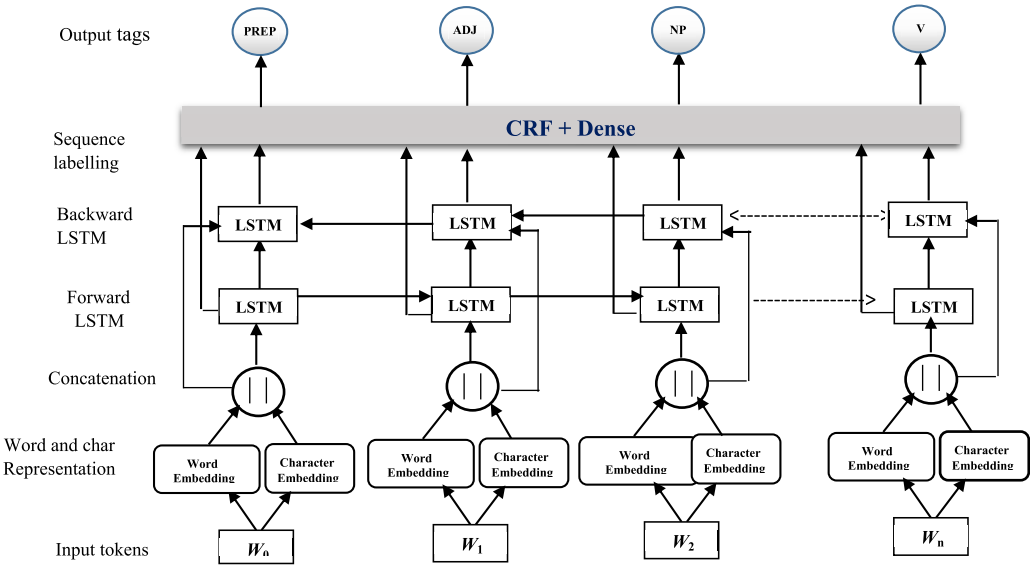


Fig. 2. Architecture of our model using BiLSTM network and CRF classifier at the top layer. At the bottom a character-based features, which is fixed sized, are derived using CNN as indicated in Fig. 1 in this paper. The model receipts an input sequence W_i (each words in a sentence words) with target sequence y (POS tags).

$$s_{CRF}(X, y) = \sum_{i=0}^{N-1} w_i . v_i + w'[y_{i-1}, y_i] \quad (4)$$

where $\boldsymbol{v_i}$ is the last dense layer output label for the ith word, and w_i is the CRF weight vector corresponding to the ith word of the same dimension as $\boldsymbol{v_i}$. $\boldsymbol{w_i}$. $\boldsymbol{v_i}$ is their dot product. $\boldsymbol{w'[y_{i-1}, y_i]}$ is the weight of the transition from the tag $\boldsymbol{y_{i-1}}$ to $\boldsymbol{y_i}$. Where $\boldsymbol{v_i}$ is the last dense layer output label for the ith word, and w_i is the CRF weight vector corresponding to the ith word of the same dimension as $\boldsymbol{v_i}$. The unknown parameters of the network are the weights and biases of the LSTM network. And the weights $\boldsymbol{w_i}$ and $\boldsymbol{w'}$ $[\mathbf{y_{i-1}}, \mathbf{y_i}]$ can be the transition weight from the tag $\boldsymbol{y_{i-1}}$ to $\boldsymbol{y_i}$.

5. Experimental design

5.1. The corpus

Amharic is among the under-resourced languages. This is a critical hindering matter that makes POS tagging in Amharic unable to achieve maximum accuracy compared to other NLP resource-rich languages. Owing to this, we tried to comprise a corpus from two different sources. The first one has been compiled from ELRC, which is a News text, manually annotated with 31 POS tags consisting of about 210 K tokens [13]. The second corpus is the religious corpus manually annotated with 62 POS tags by Gashaw [36]. The later corpus tagging strategy was by taking ELRC as a baseline so that the tagset is extended from 31 to 62. This is because they considered the plural formation of words that didn't take into consideration in the ELRC tagset. They add "S" for nouns with plural numbers (NS). Moreover, they used prepositions and conjunction with adverbs, ADVP and ADVPC. However, in the current study, we didn't consider nouns with plural and adverbs with prepositions and conjunction forms of tags. Hence, we retagged the second corpus by replacing ADVPC with ADVP, ADVPP with ADVP, NS with N and NP with N, making it have an equal number of tags with the ELRC corpus. In addition, we made rigorous corrections for the problem that was associated with tagging inconsistency; for instance, some tokens have received multiple tags under the same conditions. Then, we combined both corpora for having a diversified and large dataset for the development of our model. As Table 1 shows the noun class and verb class tags are most frequent in the corpus.

Finally, our corpus has ended up with 16,451 sentences (about 321 K tokens). During experimentation, 80% of the total corpus is allocated for training and validation purposes. The detail of the dataset is provided in Table 2.

5.2. CRF based experiment

Different features, which we manually crafted such as contextual, morphological, orthographic, and word position, are explored thoroughly to observe the most appropriate combination for Amharic POS tagging using CRF. We have also analyzed the performance of POS tagging to determine the degree to which the performance is reduced due to various features, especially context and morphology information. In the preliminary experiment, the baseline tagger is implemented, In this case, we assumed that the model will predict the most probable category using the probabilities of classes learned from the dataset and the common POS tag found in the training set. A 10-fold cross-validation is applied to get dependable accuracy.

5.2.1. Experimental results and discussion

The outcomes of CRF based model with the best combination of features are presented in Table 3. The baseline experiment exhibited satisfactory performance, yet it is not as exceptional as the other contemporary POS taggers presented in the literature. Consequently, we continued the experiment by utilizing an alternative set of features until we reached the highest performance. We performed experiments incorporating the morphological-based features of each term to measure the impact of the affix. The varying size of the prefix and suffix of the current word was taken into account to analyze their effect on the model's performance. The model has shown 5.29% enhancement of the overall F1-score over the baseline experiment while using length 2 of prefix and length 3 of suffix. Based on this, we have comprehended that the decline in POS tagging accuracy by 5.29% is due to the highly morphological nature of the language. In contrast, the contextual feature-based taggers achieved an overall F-score 83.77% that surpassed the baseline model by 15.29%. The optimal performance was achieved through a composition of features that exceeded the baseline with morphology by 10.13%. These results suggest that when using CRF algorithm, windows based information is crucial for Amharic POS

Table 1
Frequency of top eight tags from two classes in the compiled dataset.

S.No.	Tags	Frequency	
1.	N	52771	Total noun class tags
2.	NC	9227	96504
3.	NP	32456	
4.	NPC	2050	
5.	V	24212	Total verb class tags
6.	VN	5738	61471
7.	VP	21925	
8.	VREL	9596	

Table 2
Statistics of our dataset.

Dataset		
Training	# sentence	13160
	# Token	256887
Development	# sentence	1721
	# Token	33006
Testing	# sentence	1570
	# Token	31216 (10.7% unknown)

Table 3
Overall performance of the CRF based tagger.

Feature set	Window size	F1-score (%)
CRF baseline model	–	68.35
CRF baseline + Morphological	Suffix [3], prefix [3]	71.2
	Suffix [3], prefix [2]	73.64
	Suffix [2], prefix [2]	73.43
	Suffix [2], prefix [3]	73.25
CRF baseline + Context word	[-1, 1]	82.78
	[-1, 2]	82.14
	[-1, 3]	80.28
	[-2, 1]	83.5
	[-2, 2]	83.77
	[-2, −3]	81.66
	[-3, 2]	83.1
CRF baseline + Contextual [-2, 2] + Morphological [prefix [3], suffix [2]]		90.58
CRF baseline + Contextual [-2, 2] + Morphological [prefix [3], suffix [2]] + Position		91.14
CRF baseline + Contextual [-2, 2] + Morphological [prefix [3], suffix [2]] + Position + Orthographic		**94.08**

tagging. Regarding the window size of the context information, as we can see from Table 3, window size two from both sides of the current word outperformed window size one and window size three.

Succeeding this, only 0.56% enhancement was noticed upon amalgamating the baseline, morphological, and contextual features with positional features. This demonstrates that the position feature does not significantly impact the tagger's performance. On the other hand, when combining orthographic features with the baseline, contextual, morphological, and positional features, a substantial increase of 2.94% has been found on the tagger, indicating the vital role of adding the orthographic feature for identifying a word's class.

Moreover, we also examined the trade-off between training data size and the algorithm's inference capability. Our experiments involved augmenting the data size by 40 k at each step, commencing from 80 k to 320 K. The experiment result revealed that the CRF-based tagger could show better performance with an increase in training data size, implying a strong connexion between the data size and the CRF model's performance, where a larger size leads to better performance.

Generally, the best performance for this model would bethe one that shows the best F1 score among the results. This occurs with the subsequent feature set: F = {$Word_{i-2}$, $Word_{i-1}$, $Word_i$, $Word_{i+1}$, $Word_{i+2}$, 3 characters of the prefix, 2 characters of the suffix, and orthography (such as "is_symbol" and "is_digit")}. However, based on our experiment results, morphology and context information took the major share in improving the F1-score of the model; thus, both are indispensable for POS tagging tasks in Amharic. The experimental result with these feature combinations gave the Recall of 94.88%, Precision of 93.3%, and F-measure of 94.08%. Note that the features we used in this experiment are well-known in a different study, and there are no distinctive features we applied here. However, we carefully analyzed each feature to ascertain its effect and kept the model from misleading during learning. For example, we investigate different window sizes of context words; we also carefully investigate the different lengths of prefixes and suffixes of the current word. Generally, it can be said that both language-dependent and independent features are the backbone for building a high-performance POS tagger using CRF model. However, we learned that the language-independent feature was more determinant in our study.

5.3. Deep neural network-based experiment

To investigate the effectiveness of our proposed model, several experiments were performed to gauge the accomplishment of diverse DNN models. We also analyzed the most compelling features that are either at the word level alone or in a combination of both word and character levels. Besides, we also analyzed the effect of the CRF classifier at the output layer in place of using Softmax. Keras API was employed to build the seq2seq network since they can create complex models using Tensorflow in the backend.

5.3.1. Hyperparameter setting

The hyperparameters and their values involved in this study are almost similar to the hyperparameters used in Ref. [37]. We mainly

focus on tuning the embedding size, learning rate, epoch, and dropout parameters, and let others as a fixed value throughout the model training. As presented in Table 4 we fine-tune the initial values for each parameter, modifying them during gradient updates of the model by back-propagating gradients. To determine the best hyperparameters value, we utilize the development set, and the final results are obtained from the test set. At this point, we adjust the model hyperparameters by assessing the performance across every POS category in the development set.

5.3.2. Model training

In the beginning, the cleaned and fully tagged Amharic POS tagged dataset is loaded by using Pands library. Subsequently, during pre-processing, tasks such as tokenization, tagging, generating X and Y vectors, and padding sequences.

In this step, by using Tokenizer (.) function from Keras library, each word in the entire corpus is encoded by giving a blind unique id. Similarly, each character is also assigned an integer value, and each sequence of characters is encoded as an integer sequence. As we also mentioned in the previous section, we used word embedding for input sequences (X). Conversely, we possess the Y matrix tag/output data. We hold 31 POS tags in this case, considering every one of them as a class, and each POS tag is transformed into a one-hot encoding comprising 31 elements.

to_categorical function from Keras' is used to one-hot encode Y = to_categorical (Y) and X = to_categorical (X). Likewise, it necessitates a unique hot encoder for every character. This implies that each character becomes a vector as long as the vocabulary with 1 is marked for the specific character. A 1D-CNN is utilized to obtain a numerical portrayal of words by analyzing their character-level structures, with an input dimension of m = 30 for each word.

Subsequently, the input characteristic matrix, which has dimensions of m x 100, is fed into a dropout layer with a dropout rate of 0.3. The purpose of the dropout layer is to minimize the issue of overfitting. Four convolution filters were applied, with kernel sizes of 3, 5, and 7, a stride of 1, and padding of 0. Each convolutional filter provides an output vector of 30 -dimensions. Consequently, the overall size of the character-dependent word vector is 100. Afterwards, extracted features are provided as input to the pooling layer and then produced vector outputs 3 (1×90). Following this, it is important to know the size of the vocabulary. This could be obtained as the size of the dictionary mapping. The sentences and each word in the corpus are not of the same length. Hence, before we fed the input to the network, we fixed the length of the sentences and characters by padding 0's for sequences less than 100, and 30 characters in a word in length by using keras' pad_sequences (.) function. Once the above tasks were accomplished, word embedding followed, and we have gone to word2vec model for word embedding. These vectors are then concatenated that have word-level, sub-word-level, and character-level information, and fed into the subsequent BiLSTM unit. However, before starting the modelling process, it is essential to verify that the data dimensions conform to the requirements of an LSTM. Typically, at this juncture, the shape of X: (#instances, #timesteps, #attributes) and the shape of Y: (#instances, #timestamps, #attributes). Another important stage is the modelling part. The model has a BiLSTM hidden layer with 64 LSTM cells. Finally, the pre-processed data as a whole is fed into the BiLSTM network and then the concatenated hidden states of BiLSTM to the CRF layer.

5.4. Results and discussion

We have performed a comparison based on the experimental results obtained from LSTM, BiLSTM, and BiLSTM with CNN jointly on POS tagging of Amharic text. According to the results shown in Table 5, the LSTM architecture, which uses word embedding only, performs poorly, with an accuracy of 89.7%. However, both BiLSTM and BiLSTM-CRF using word embedding obtain better performance than others, with an accuracy of 92.83% and 94.61%, respectively. Merely utilizing the word embeddings vector while conducting training didn't yield encouraging results. As the best solution suggested for this problem is adding character-based word embedding to the word embedding. While considering both word and character embedding jointly, both models have shown a better result than the one using only word embedding. Comparatively, The CNN-BiLSTM-CRF architecture exhibited slightly superior performance compared to the BiLSTM-CRF architecture, achieving an accuracy score of 97.23% and BiLSTM is about 2.03% behind. When we consider BiLSTM for character-level learning as which was proposed by Ling et al. [38], our experiment revealed that it doesn't perform better than CNN though being more computationally expensive to train.

Table 4
Hyper-parameters and their value.

Layer	Hyper-parameter	Range	Final value
CNN	Pool size	–	Global-max
	Max. Char. len.	–	30
	Window size	–	3
	Number of filters	–	30
LSTM	State size	[100–500]	200
Dropout	Dropout rate	[0.25–0.5]	0.3
	Word emb. dim	[50 or 100]	100
	Batch size	32 or 64	32
	Epoch	[10–40]	20
	Initial learning rate	$[10^{-6} - 10^{-1}]$	0.01
	Max. Sent. length	–	100
	optimizer	–	Adam

Table 5
Tagging performance of DNN based models on development and test set.

Model	Feature	Accuracy (%)	
		Dev't set	Test set
LSTM-Softmax	Word embedding	92.38	89.7
BiLSTM-Softmax		99.29	92.83
BiLSTM-CRF		99.13	94.61
LSTM-Softmax	Character and word embedding	97.21	94.21
BiLSTM-Softmax		99.02	95.2
BiLSTM-CRF		99.89	96.82
CNN-BiLSTM-CRF		99.98	**97.23**

Considering the tagging accuracy as the percentage of correctly assigned tags, we have also gauged the performance of the best scorer model with the known and unknown words. It is important to determine how the model would perform with words that were not included in the training dataset. For CNN-BiLSTM-CRF run, Table 6 shows the percentage of seen words, the number of tokens in the test set, the number of tokens correctly tagged and the percentage of accuracy for that run. Generally, the precision for familiar terms is considerably greater than that for unfamiliar terms (97.01% vs. 77.24%). This indicates a discrepancy of 19.77% in accuracy between recognized and unrecognized words.

It is also observed that using the CRF classifier at the output layer could lead to improving the performance of CNN-LSTM based tagger by 1.62%, which is a notable improvement, and escalated the previous latest accuracy to 97.23%. This designates that models based on character-level possess enough intricacy to acquire the adaptable patterns of morphology and lexical.

Certainly, it was expected such improvement from CNN-LSTM-CRF based tagger, since our task is attributed to a sequence labelling problem. On top of this, as depicted in Fig. 3, we also measured the number of epochs by varying the value in the interval from 5 to 20. The training required less than 10 epochs to converge. Thus, it is conceivable that 10 epochs are optimal for BiLSTM-based tagger for successful learning. Moreover, the loss value drops instantly when the epoch comes to about 3, then oscillates at the bottom. Hence, we can decide that while the number of epochs increases, the validation loss decreases.

As can be seen in Fig. 4, with a minimal value of learning rate, that is less than 1E-4 (0.0001), the model couldn't able to discover patterns but consecutively improved as the value get higher than this point. Finally, at a 0.01 learning rate value, the model became super in learning and prediction. We obtained significant improvements in model performance after using dropout. In addition, the total number of parameters generated by the CNN-BiLSTM-CRF based model is 1,823,729. Finally, we have seen that all deep sequence learners perform equally well when a considerable size of training data was provided.

6. Discussion based on CRF and CNN-BiLSTM-CRF model

So far, we have evaluated four POS tagger models: CRF, LSTM, BiLSTM, and CNN-BiLSTM. All models have been trained on a similar dataset so that the model's result can be comparable. LSTM with word embedding is the least performed model. However, it presents a comparable value with the CRF when trained with both level features but is worse than BiLSTM. We repeatedly obtained a better accuracy with the BiLSTM model than the CRF framework with a similar training set but with different feature sets. The BiLSTM using only word embedding performs better than LSTM with character and word level embedding. There are 3.6% numerical differences between the CRF and CNN-BiLSTM models.

This proves our assumption that DNN is the contemporary approach for POS tagging in Amharic and is also based on the studies [5, 6] discussed in section one. Yet it is a numerical difference we have shown, but it should be proved whether the two models essentially differ or not, using statistical analysis. As the purpose was to establish whether there is a real performance difference between the CRF and CNN-BiLSTM models, an independent samples ***t-test*** involving an attributed factor was considered for the appropriate statistic to test our assumptions. Here, the data size and accuracy were taken as a statistical variable because both the learning ability of models was profoundly affected by the amount of data. This revealed that the data size and model performance are highly correlated, their correlation value, $r = 0.902$.

The distribution of data size was sufficiently normal for the purpose of conducting a *t-test*. As provided in Table 7, the test result revealed that *t-value is 0.0021*. A 5% level of significance (i.e., allowing only a 5% of chance that the variance between the two model means is due to chance) was set for the significance test. The *t-value* is less than the set level of significance of 0.05, and this means the variances are statistically significantly different. Therefore, we can declare that the CNN-BiLSTM-CRF model can perform more effectively and outperform CRF in POS tagging for Amharic text.

Regarding model robustness, our dataset has highly imbalanced labels and many ambiguous labels, especially in the religious

Table 6
Tagging performance of the proposed model on Known and Unknown tokens.

Testing Set	No. Tokens	Correct	Accuracy
Known tokens	27876 (89.3%)	27043	97.01%
Unknown tokens	3340 (10.7%)	2580	77.24%

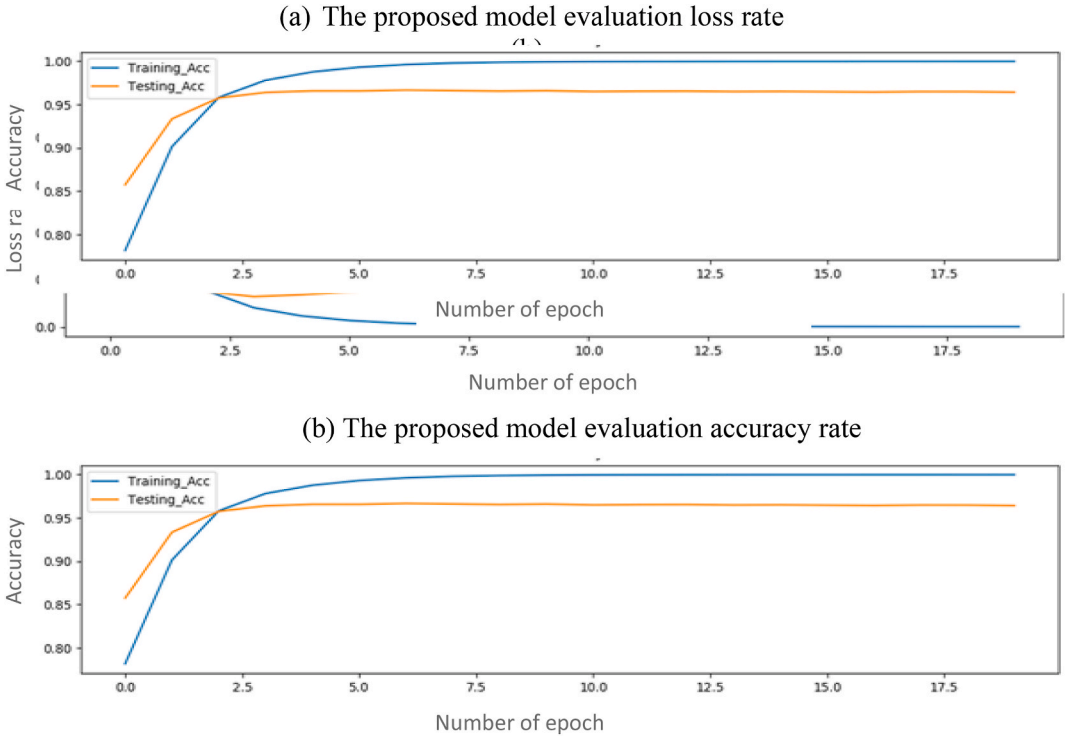


Fig. 3. The chart illustrates the accuracy and loss of the model in the training and testing data sets by utilizing the CNN-BiLSTM-CRF. In the accuracy model, preliminary validation accuracy is 0.86 but after two epochs the validation accuracy instantly get increases to 0.95. With the same token, the preliminary validation loss is about 0.5 but after two epochs the loss declines to below 0.2. This indicates a complimentary trend towards enhancing accuracy and dropping loss.

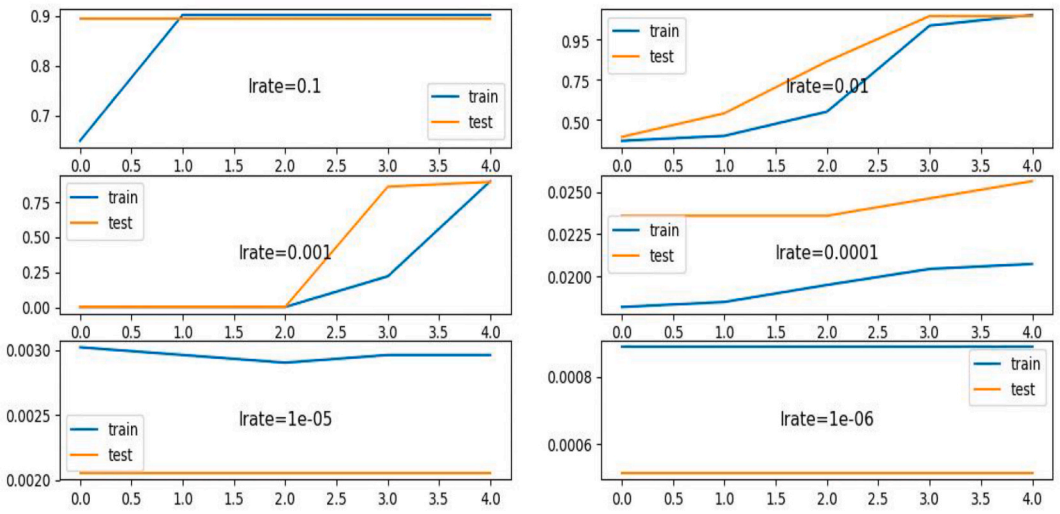


Fig. 4. Graph representing train and test accuracy over epochs for a suite of learning rates of the proposed model on the POS tagging problem.

Table 7
Descriptive Statistics for the performance of models by data size (Indp. Sample *t*-Test).

Models	Mean	Std, dev.	t-value
CRF	85.71	6.9	0.0021
CNN-BiLSTM-CRF	92.12	3.17	

corpus. While we trained the CRF framework on such a dataset without making any corrections, its loss function was high. Congruently, the performance was significantly degraded. But, the level of this problem realized in CNN-BiLSTM-CRF based model became lower; proving that the CNN-BiLSTM-CRF based model is more robust, i.e. it is insignificantly affected by the label imbalance and noisy data. This also led us to conclude that the DNN, particularly CNN-BiLSTM-CRF, is the best approach for POS tagging in Amharic.

6.1. Performance comparison with related works

To show the proposed method's effectiveness, a performance comparison between our best-performer taggers with each other and other existing state-of-the-art taggers has been made. As they are listed in Table 8, very recently, Akbik et al. [9] proposed the best tagger with an accuracy of 97.85%. In their work combination of BiLSTM and CRF, and using contextual string-level embedding had selected as the best approach. The second progressive tagger was proposed by Huang et al. [4] and they obtained 97.55% accuracy by using BiLSTM-CRF. A couple of years back, Santos and Zadrozny [26] proposed a tagger that achieved 97.32% accuracy by using character embedding. In the same study but reported on the Portuguese language, the highest F1 score of 97.47% using BiLSTM with word and character embedding had achieved. Moreover, 97.6% accuracy was achieved in a morphologically rich language, Indonesia, by Kemal and Aji [39]. They used BiLSTM-CRF with an affix feature in addition to word and character embedding.

In our experiment, we got the highest accuracy of 97.23% with word and character embedding. To the best of the researcher's knowledge, this is the top performance ever reported in Amharic. In comparison with the [4,9], our model's accuracy is a bit lower, this can be attributed to the type of word embedding and the architecture type in character embedding we used. Most of the taggers mentioned above had used pre-trained word embedding, which has been trained on 6 billion tokens like GloVe. Furthermore, the quality of the corpus they used is also another factor; most of the works are evaluated by using the more established corpora like WSJ, and Penne Treebank. Although our tagger's performance lags behind state-of-the-art taggers, our experiment proves the efficiency of a DNN using minimal feature engineering which can provide a unified solution for POS tagging and other sequence labelling tasks in Amharic. When we compared our proposed model with the only work developed for Amharic by using a deep learning approach (BiLSTM) [10], our model's performance is much superior, surpassed by 3.56%. We recognized two major reasons why our proposed model's performance is improved than previous similar work, particularly [10]. The first one is the corpus size, in this study, a corpus which contained 321 K words is used, in contrast, they used 210 K words; the second factor is character level features experimented in this in this study helped to minimize OOV words. Generally, we have empirically proven the efficiency of DNN using minimal feature engineering that can be a unified solution for POS tagging in Amharic. But it is worth further study on how the DNN can be more practical and provided to public users for Amharic POS tagging.

6.2. Analysis of POS tagging errors

It is helpful to scrutinize the failure cases of our model to improve the performance of our proposed tagger further. We have conducted an error analysis, taking sample errors from the CNN-BiLSTM-CRF model. The tagger predicts most tags seamlessly, yet errors are observed in some tags. Mainly, the nouns, adjective and adverb tag class's confusion accounted for the major error, for instance, NP tag was confused 347 times with ADJ tag, 50 times with ADV tag. Secondly, VP and ADJ where VP tag is confused 103 times with ADJ tag, and ADV with ADJP and ADJ, where it is confused 24 and 42 times with ADJP and ADJ tags respectively. Moreover, more than 18% of the errors in BiLSTM-CRF resulted from considering non-nouns and their options (i.e., NP, NC, and NPC) as noun families.

While we were closely looking at such conflicts in tagging (or errors), our experiments revealed that such errors have arisen from three major sources. The first source of error might be systematic in tagging, which means that, given a particular word in a particular context, the tagger will always tag it incorrectly. This behavior is attributed to tagging errors in the corpus on which the model was trained. This is the gold standard inconsistent. This inconsistency leads the model unable to distinguish the tags properly. For example, expressions like the be2xih/by two thousand, are inconsistent, at some point, they are tagged as NUMCR and in others as NUMP. Also xih/thousand is tagged as "NUMCR" and "N" in another position, thus it is very realistic for a POS tagger to predict that it could be a noun, as it did (see Fig. 5 as evidence). Moreover, the token "200xih/200shih" shih means thousand should have been tagged as "NUMP". Out of 12478 occurrences of the token in the corpus, 548 times had been tagged as "NUMCR" and "NUMP" the remaining 10739 times. Fortunately, BiLSTM based tagger scored sound accuracy on "NUMP" tag, but CRF based tagger did not, which implies that DNN model is robust to noise in POS tagging in Amharic.

The second source of error was the context problem, meaning that certain words appeared in the dataset frequently, but never had

Table 8
Comparison of our models with state-of-the-art taggers.

Technique	Dataset	Accuracy
CRF (This paper)	–	94.08%
CNN-BiLSTM-CRF (This paper)	–	97.23%
BiLSTM [10]	ELRC	93.67%
BiLSTM-CRF [4]	Penn TreeBank	97.55%
MLP with character Embedding [26]	WSJ Portuguese language	97.32%
BiLSTM-CRF [9]	Germen and English	97.85%
BiLSTM-CRF [39]	Indonesia	97.6%

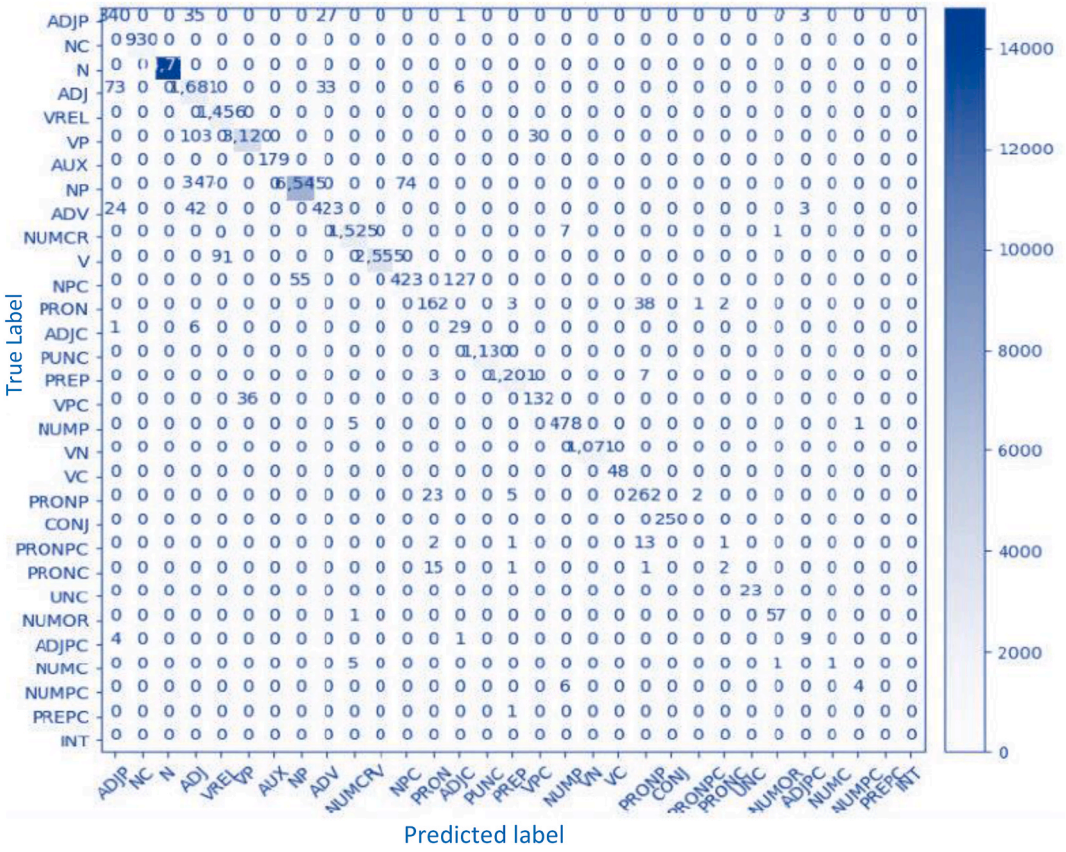


Fig. 5. A confusion matrix of each tag in the test set of CNN-BiLSTM-CRF.

the tag that currently has with this context. Our experiment revealed that the model mainly relies only on context features, but often the context information is ambiguous. Consequently, the tagger faced a strong challenge to tag these terms correctly. For instance, as you can see in Fig. 5 the term "Akababi" is certainly a Noun; nonetheless, it appeared in training data many times tagged as an adjective in every context. This problem can be reduced by adding much syntax/semantics/discourse and using pre-trained word embedding, the one who trained with a diversified domain other than the News corpus.

One more source of error was from unknown words (OOV) and the lexical problem. These problems led the taggers to commit an error and accounted for 41% out of the whole error on CRF based model and 28% on CNN-BiLSTM based tagger. This implies that the character level embedding approach has been provided as a solution for unknown words. Furthermore, combining prefix and suffix information embedding may minimize this problem. Also, developing pre-trained embeddings for Amharic could be provided as the best means to handle unknown words.

7. Conclusion and future directions

The POS taggers discussed in this study are straightforward and state-of-the-art for automatic Amharic text POS tagging tasks. We performed a comparison between CRF and DNN (LSTM, BiLSTM and CNN- BiLSTM-CRF) models. We have developed and tested each model using a similar dataset that is tagged manually. For the CRF model development, we have studied the exact nature of various features, these are mainly contextual, morphological, and orthographic. And several experiments have been conducted to comprehend the best feature set. Moreover in DNN experiment, various issues were analyzed, (1) the impact of features at the word level and character level; (2) the architecture such as LSTM or BiLSTM or combination of BiLSTM with CNN; (3) the classifier at the output layers, for instance, softmax or CRF; (4) giving a depth knowledge of validation filters in CNN, exploration of different epochs, learning rate, dropout in the models training. Thus, this gives a better understanding of our proposed model. Accordingly, the CNN-BiLSTM-CRF model performed a substantial result and is comparable with the contemporary taggers in English and other languages. As the one contribution of this study, the joint embedding developed here is not limited to the Amharic language; it can also be applied to agglutinative Ethiopian languages, such as Afan Oromo as it could be benefited if using character-level embedding.

There is still room for improvement in the proposed system; for instance, our model demands a large amount of manually tagged data to obtain dependable achievements. Nevertheless, it is tedious for tagging big training data manually. Therefore, there should be a way to automatically produce training data as the best solution. However, using machine-labelled data merely does not guarantee to

development of a competent model due to automated tagging errors. Thus, to confiscate such errors, studies are highly recommended to use a transfer learning approach.

Author contribution statement

The authors confirm their contribution to the paper as follows:

Sintayehu Hirpassa (PhD) and G.S. Lehal (Professor): Conceived and designed the experiments; Performed the experiments; Analyzed and interpreted the data; Contributed reagents, materials, analysis tools or data; Wrote the paper.

Data availability statement

Data will be made available on request.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] H. Daniel, Jurafsky. Part-of-Speech tagging, Speech Language Process 1 (2) (2019) 1–28.
[2] M. Banko, R.C. Moore, Part of speech tagging in context, in: *Proceedings of the 20th International Conference on Computational Linguistics COLING*, 2004, pp. 556–561.
[3] Alex Graves, Supervised sequence labelling with RNN, Language Technology 385 (9) (2019) 45–52.
[4] Z. Huang, W. Xu, K. Yu, Bidirectional LSTM-CRF Models for Sequence Tagging, 2015. *ArXiv*, abs/1508.0.
[5] M. Poel, et al., A Neural network based Dutch part of speech: tagger, in: Artificial Intelligence Conference, 2008, pp. 217–224.
[6] R. Alharbi, et al., Part-of-speech tagging for Arabic Gulf dialect using Bi-LSTM, in: 11th International Conference on Language Resources and Evaluation, 2019, pp. 3925–3932.
[7] Z. Wang, et al., A survey of deep neural network architectures and their applications,", Neurocomputing 234 (5) (2017) 11–26.
[8] H. Salehinejad, et al., Recent Advances in Recurrent Neural Networks, ArXiv, 2018, pp. 1–21.
[9] Akbik Alan, et al., Contextual string embeddings for sequence labeling, in: International Conference on Computational Linguistics, 2018, pp. 1638–1649.
[10] M. Argaw, A, Amharic Parts-Of-Speech Tagger Using Neural Word Embeddings as Features Amharic POS Tagger Using Neural Word Embeddings as Features, AAU, 2019.
[11] D.G. Anastasye, et al., Improving part-of-speech tagging via multi-task learning and character-level word representations, Computer. Linguistic Intellectual. Technology 17 (5) (2018) 14–27.
[12] "Amharic language - dialects & structure." [online], Available: https://www.mustgo.com/world languages/Amharic/.
[13] G. Demeke, M. Getachew, Manual annotation of Amharic news items with part-of-speech tags and its challenges, Ethiopian Language Research Center Paper 2 (2006) 1–16.
[14] M. Gasser, HornMorpho: a system for morphological processing of Amharic, Oromo, and Tigrinya, in: Conference on Human Language Technology for Development, 2011, pp. 1–55.
[15] N. Alemayehu, P. Willett, Stemming of Amharic words for information retrieval, Literature Linguistic Computer 17 (1) (2002) 1–17.
[16] B. Gambäck, A.A. Argaw, L. Asker, Methods for Amharic part-of-speech tagging, in: First Workshop on Language Technologies for African Languages, 2009, pp. 104–111.
[17] Mesfin Getachew, "POS TAGGING," Addis Ababa University, 2001.
[18] S.F. Adafre, POS tagging for Amharic using conditional random fields, in: Proceedings ACL Workshop on Computational Approaches to Semitic langEstimation of Conditional Probabilities with Deciuages, 2005, pp. 47–54.
[19] G. Binyam, Part of speech tagging for Amharic, in: Proceedings of Natural Language Processing & Human Language Technology, 2010, pp. 1–20.
[20] M.Y. Tachbelie, W. Menzel, Amharic part-of-speech tagger for factored language modeling, in: International Conference Recent Advances in Natural Language Processing, RANLP, 2009, pp. 428–433.
[21] Y.T. Martha, Tet al. Part-of-Speech tagging for under-resourced and morphologically rich languages – the case of Amharic, in: Conference on Human Language Technology for Development, 2011, pp. 50–55.
[22] J. Lafferty, A. Mccallum, Conditional random fields, in: International Conference on Machine Learning, ICML), 2001, pp. 282–289.
[23] E. Fosler-Lussier, R. Prabhavalkar, Conditional random fields in speech, audio, and language processing, in: Proc. IEEE 101, 2013, pp. 1054–1075.
[24] T. Young, et al., Recent trends in deep learning based natural language processing, IEEE Computer Intelligence Management 13 (3) (2018) 55–75.
[25] C.N. Dos Santos, B. Zadrozny, Learning character-level representations for part-of-speech tagging learning,", Journal of machine learning research 32 (2017) 26–37.
[26] J.P. Chiu, E. Nichols, Named entity recognition with BiLSTM-CNNs, Transaction. Association. Computer. Linguistic 4 (2016) 357–370.
[27] F. Gers, Long short-term memory in recurrent neural networks, Neural Computing 12 (10) (2001) 2451–2471.
[28] R. Rojas. The backpropagation algorithm. *In* Neural Networks, Verlag, Berlin: Springer, 1196.
[29] Y. Hu, A. Huber, S.-C. Liu, Overcoming the Vanishing Gradient Problem in Plain Recurrent Networks, ArXiv, 2018, pp. 1–20.
[30] R. Pascanu, et al., On the difficulty of training recurrent neural networks, in: 30th International Conference on Machine Learning, ICML, 2013, pp. 2347–2355.
[31] X. Ma, E. Hovy, End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF, in: Proceedings of the Association for Computational Linguistics 4, 2016, pp. 1064–1074.
[32] L.C. Molina, et al., Feature selection algorithms: a survey and experimental evaluation, in: International Conference on Data Mining, 2002, pp. 306–313.
[33] Y. Li, T. Yang, in: S. S (Ed.), Word Embedding for Understanding Natural Language : A Survey. *Studies in Big Data*, Springer, Cham, 2018, pp. 83–104.
[34] R. Collobert, et al., Natural Language processing: almost from scratch, Journal of Machine Learn. Research 12 (2011) 2493–2537.
[35] G. Ibrahim, et al., Machine Learning Approaches for Amharic Parts-Of-Speech Tagging, Proceeding of ICON, 2018, pp. 74–79.
[36] N. Reimers, I. Gurevych, Optimal hyperparameters for deep LSTM-networks for sequence labeling tasks, computer technology 17 (3) (2017) 25–40.
[37] W. Ling, et al., Word representation finding function in form : compositional character models for open vocabulary word representation, in: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, 2015, pp. 1520–1530.

[38] K. Kurniawan, A.F. Aji, Toward a Standardized and More Accurate Indonesian POS Tagging, vol. 1809, ArXiv, 2019, pp. 303–307.
[39] A. Krizhevsky, I. Sutskever, ImageNet classification with deep convolutional neural networks, in: Proceedings of Advances in Neural Information Processing Systems, 2012, pp. 257–268.
[40] Q. Nguyen, K. Verspoor, From POS tagging to dependency parsing for biomedical event extraction, BMC Bioinf. 20 (72) (2019) 195–258.