Design and Develop Automatic Parts of Speech Tagger for Siltigna Language

Conference Paper · November 2024

DOI: 10.1109/ICT4DA62874.2024.10777240

5 authors, including:

Michael Melese Woldeyohannis
Addis Ababa University
34 PUBLICATIONS **244** CITATIONS
SEE PROFILE

Eyobed Birhanu Paulos
Wolaita Sodo University
3 PUBLICATIONS **15** CITATIONS
SEE PROFILE

Mesay Gemeda Yigezu
Interino del Centro de Investigación en Computación Instituto Politécnico Nacional
20 PUBLICATIONS **52** CITATIONS
SEE PROFILE

DESIGN AND DEVELOP AUTOMATIC PARTS OF SPEECH TAGGER FOR SILTIGNA LANGUAGE USING DEEP LEARNING ALGORITHM

Biniyam Sime Dana
School of Computing
Hawassa University,Ethiopia
binyam.sime@wsu.edu.et

Michael Melese Woldeyohannis
School of Information Science, Addis Ababa University
Addis Ababa, Ethiopia
michael.melese@aau.edu.et

Seiyfedin Mohammed Yesuf
School of Informatics, Wachemo University
Hossana, Ethiopia
moahmmedseiyfedin@gmail.com

Eyobed Birhanu Paulos
School of Informatics
Wolaita Sodo University, Ethiopia
eyobed.birhanu@wsu.edu.et

Mesay Gemeda Yigezu
Instituto Politécnico Nacional
Mexico
messay.gemeda@gmail.com

***Abstract*—The purpose of this study is to design and develop automatic part-of-speech taggers for the Siltigna language using the deep learning algorithm. Part of speech tagging is the process of assigning words within sentences to the appropriate word categories. In most languages POS tagger is utilized as a pre-processing component in several NLP applications. Considering that several part-of-speech taggers are developed for various foreign and local languages. However, those POST cannot be applied directly with Siltigna due to morphological difference between the languages. To build tagging modelsvarious deep learning algorithms such as Standard RNN, GRU, LSTM, and BiLSTM were implemented. For training and testing purposes, 3500 sentences were collected from different sources, such as Siltigna text books, annual and monthly reports from the tourism office, and Siltigna newspapers, in sound format, then converted to text format. Some proverbs, which are written in Siltigna, were also collected to balance the sample corpus, and 22 sets of tags were identified for tagging purposes. The POST tagger is tested using a percentage spilled mechanism. The experimental result indicates that for simple RNN, LSTM, GRU and bidirectional LSTM was 92. 49 %, 92. 55 %, 92. 61 % and 93. 05 % test accuracy, respectively. Based on the result obtained from the accuracy test, conclusions and recommendations are forwarded.**

***Index Terms*—Siltigna language, POS tagger, NLP, RNNs, neural word embedding**

I. Introduction

Natural language is any language that humans use to communicate with one another [1]. As stated in [2], in addition to being the most important form of human interaction, natural language has been used historically to preserve cultural achievements and pass them down from one generation to the next. NLP is a computerized way of analyzing text based on a set of concepts and technological advances. It also pertains to the use of computers to process and comprehend natural language. [3]–[5]. NLP is a tool for carrying out various activities or applications [6]. Among those tasks, part of speech (POS) tagging is one of the activities carried out by NLP. Part of speech labeling is the technique of allocating words within sentences to their relevant parts of speech or word groups [7]. NLP applications such as speech recognition, speech synthesis, information retrieval, word sense disambiguation, and machine translation rely heavily on it [8]. Advanced NLP applications in Siltigna or any other language require the use of POS tagging. POS tagger is currently being developed locally and internationally for various languages, and it is still a hot topic for study for different languages. Since many indigenous African languages, including the Siltigna language, are under-resourced, there are few computational linguistic tools or corpora (such as lexica, taggers, parsers, or tree banks) available [7]. POS taggers have been developed for local languages such as Amharic, Awgni, Afan Oromo, Wolaita, Tigrigna, and others [4], [9]–[12]. However, due to the language difference, the part-of-speech taggers developed for a specific language are not directly applied to other languages [4].

Therefore, the aim of this study is to design and develop POS tagger for Siltigna language so as to construct the base for future scholars who have an interest in the area of NLP applications.

II. Parts of speech tagging

Parts-of-speech Tagging is the technique of allocating words within sentences to their relevant parts of speech or

word groups [7]. As stated in [11], POS tagging assigns whether a given word is a noun, adjective, verb, etc. As we have discussed in the introduction part, POST are mainly used in order to solve ambiguity of languages. Ambiguous words are common problems in the Siltigna language. For instance,

ቆጨዮ የብሌን በሂም። (tortoise is wild animal).**ጀማል ህንጠ ቆጨዮ ።** (Jemal cuts the tree). From the above two sentences, the words are conveying different meanings based on their contextual structure. In the first sentence, the word **“ቆጨዮ”** means "tortoise," which is used as a noun, whereas in the second sentence, the meaning of the word is “cut” and used as the verb.

A. Approaches of parts of speech tagging

A word's part-of-speech cannot be identified simply by viewing the word from a given corpus [13]. For instance, in the above example, the word **“ቆጨዮ”** has different parts of speech in different contexts; it is a verb in the phrase and a noun in the other phrase. So a simple corpus lookup or a morphological analysis produces many words that are ambiguous, especially nouns and verbs. Therefore, to solve such kinds of problems, automatic parts of speech tagger are the best tools. So, there are different approaches that are employed to develop automatic POS taggers. Those approaches are used to annotate words automatically with their parts of speech tags from the given corpus. The most common ones are rule-based, stochastic or probabilistic-based, artificial neural networks, hybrid, and deep learning approaches.

III. Related Works

Since there are different studies for parts of speech tagging that have been conducted in foreign and local languages. Among them, let us see some of the works that are related to our works on Table I below.

IV. The structure of Siltigna word classes

A. The writing system of Siltigna language

Each language has its own grammar rules and sentence construction techniques.Silt'e uses Ge'ez or Ethiopic characters for writing purpose since the 1980s. Although there are typically seven vowels in languages like Amharic and Tigrigna, the vowels in the Siltigna language are different from those languages. In written Siltigna language, the seven Ethiopic vowels are mapped onto the ten Siltigna language vowels as follows: the five short vowels are **ኦ** |a|, **ኡ** |u|, **ኤ** |e|, **እ** |i|, **ኦ** |o| and the five long vowels are **ኣኣ** |aa|, **ኡኡ** |uu|, **ኢኢ** |ii|, **ኦኦ** |oo|, **ኤኤ** |ee|.Only the short and long a and the short and long i have been distinguished in the alphabet. The short ‘a’ is indicated by the first order (form) of the Ethiopic script and the long ‘a’ by forth order (form). The short ‘i’ and consonant alone are indicated by sixth order (form) and the long ‘i’ is indicated by the third form. The third form is also used word finally when the word ends in ‘i’ [25]. Siltigna language has a typical collection of consonants, including the letter **ፕ** |p|.

B. Siltigna word classes

Three fundamental characteristics are typically taken into account when classifying words into their respective parts of speech. They are: the word's connotation, its physical appearance, and the context in which it appears in a sentence. These can serve as the primary criteria for classifying a specific word [23]. Because of this, traditional Siltigna grammar categorizes words into eight groups or parts of speech namely verbs, nouns, adjectives, adverbs pronouns, prepositions, conjunctions, and interjections depending on the grammar of other languages, rather than categorizing words based on Siltigna language features. The early scholars' categorization is used in this study. The reason for this preference is because the early classification is more thorough and enables the tagger to tag terms thoroughly.

C. Tags and tagset for Siltigna language

Tags are labels that provide additional information about each word in sentences [2]. A tagset is a group of identifiable tags that are employed in the creation of a research prototype. To the best of researchers knowledge there is no publicly accessible tagsets for Siltigna language. As a result the researchers uses several methods in order to identify and develop tagsets. The tag sets that are discussed below are classified as a basic class and subclasses of the basic classes. Since, nouns, pronouns, verbs, adjectives, prepositions, conjunctions and adverb are considered to be the basic word classes. In addition, numerals and punctuation are also included as basic word classes in the process of identifying the tag sets.

V. Design of the proposed Siltigna POS tagger

A. Data collection and preparation

To conduct this research, the required data was collected and pre-processed by the researcher. The researcher collected 3500 Siltigna sentences in order to prepare sample corpus from different sources in order to make the sample corpus balanced but it is in its raw form. Experts, especially linguists with experience in the topic, manually annotated the corpus that was compiled for this study. The tagging procedure is based on the determined tagset and manually tagged corpus, taking into account the contextual position of words in a phrase.

B. Proposed model framework

The study work was conducted with designing and developing parts of speech tagger model for Siltigna language. The model is trained on deep learning algorithms such as RNN, LSTM, GRU and BiLSTM using automatically generated neural word embeddings as a feature. The proposed Pos tagger system consists of a series of tasks such as tokenization, word vector generation, feature

Table I
Summary of related works

Author's	Objective	Methodology	Findings
[14][	To develop Arabic Part-of-speech Tagger	Combination of statistical and rule-based techniques was used & a corpus of 50,000 words was collected. Finally, the researcher uses percentage split mechanism to test the tagger.	statistical tagger achieved a 90% accuracy rate
[15]	To develop hybrid-based part-of-speech tagger for Turkish	Uses the combination of rule-based and probabilistic methods and to develop the model the dataset that contain 7200 sentences were used and 13 tagsets are assigned them. percentage split mechanism (83% for training and 17% for testing) was used to test the model	Experimental result shows average accuracy of 84.7% is obtained
[16]	To develops POS tagger for English language using BiLSTM	Uses BiLSTM and word embedding as features to develop the model	Tagging accuracy of 97.40% is obtained
[10]	To develop POST for Awgni language Using HMM.	Uses HMM, 23 custom tagsets & gathered 94,000 sentences.10 fold cross-validation mechanism was used.	Unigram and bi-gram taggers achieve accuracy of 93.64% and 94.77% respectively.
[17]	To develop Statistical based POS tagger for Somali language	Uses HMM, CRF, and NN, create corpus, which contain 14,369 tokens that represent 1234 Sentences & 24 tagsets. 10 fold cross-validation was used	All POS tagger scores 87.51% average accuracy
[18]	To conducted a comparison between CRF, TnT, NB taggers	Compare taggers based on HMMs such as TnT, CRF & NB. Uses existing ELRC corpus with 210K token by incorporating a manually tagged Corpus with 31tags.	CRF-based Amharic POS tagger achieved an average accuracy of 94.08%.

extraction, training deep learning algorithms that have been used for development of the final model.

1) Tokenization: The tagger starts tokenizing segmented corpus that is segmented by sentence segmenter.

2) Dictionary preparation: In this stage separation of the words from the tags was done to develop parallel corpus that contain sentences and tags as separate sequences where, the first one consists of Siltigna plain text or sequence of word and other consists of only the corresponding tags of the word sequence. After that we compute a set of unique words and tags and transform it in a list and index them in a dictionary. These dictionaries are the word vocabulary and the tag vocabulary.

3) Generating Word vectors: In order to process an input text and generate a set of vectors as an output, a two-layer neural network named Word2Vec is developed. It converts a text input into a format that a deep learning algorithm can interpret. The Word2Vec tool transforms each word into a vector of real numbers in a finite number of dimensions using raw text as its input.

4) Feature extracting: After extracting features for words of the annotated corpus, the data is given as input for a variant form of RNN.

C. Model building

Model building is implemented using variant forms of Recurrent Neural Network such as Simple RNN, LSTM, GRU and BiLSTM. The best model out of them was chosen to develop the Siltigna parts of speech tagger models.

1) Network architecture: In order to build the model with each deep learning algorithm we have consider the following layers. Embedding layer: Embedding layer is the first hidden layer of a network that computes a word vector model for our words (it converts the input words from their unique number identifiers to their word vector.

Various form of RNN: Those layers are the second and studies the pattern in the sequences presented by the data and creates feature patterns which make it easier for the fully connected layer to make a prediction on the data.

Dense Layer or Fully Connected Layer: This layer is the third layer that takes its input from the previous layer and transforms into a lower dimensional form. This lower dimensional vector is of the size of the number of tags. The output is passed through the softmax activation function which converts the vector to probability values. The tag with the highest probability is considered as the output tag.

The Neural Network is made to reduce the classification error using Categorical Cross Entropy as the Loss function and Adam with the value of 0.001 as the optimization algorithm to reduce the defined loss. For the purpose of the study we have trained each model on manually tagged Siltigna corpus with 300 epochs and batch size of 128 using the sentences which is prepared as training set. Finally using a testing dataset, the built models are evaluated to test how well the model is performed. Various metrics are used in machine learning and deep learning according to the situation which are appropriate for the given problems. For this study as performance measurement Accuracy is used and it represents proportion of accurate predictions to all other predictions.

VI. Experimental result and discussion

Three experiments are conducted in order to answer the research questions of this study with different deep

Table II
TENTATIVE POS TAG SET FOR THE SILTIGNA LANGUAGE

Basic Category	Derived Tag	Description	Example
Noun	N	Includes all types of nouns, invariant for number, gender and case	ስልጤ(silte)፣ ሰይፈድን(seiyfedin),ዝላም(rai)
	NPREP	Noun with not separated Preposition	በስልጤ(in silte)
	NC	Noun with not separated Conjunction	ከድርም(Kedir also)
Pronoun	PRO	Tag for all pronouns invariant for number, gender and case	ኡሀ(he),እሸ(she),የን(our)
	PROPREP	Pronouns with not separated Preposition	ተን(with us)
	PROC	Pronouns with non separated conjunctions	በንም(with us also)
Verb	V	Tag for all main verbs	በለ(eat),መጠ(come)
	AXU	Auxiliary verbs in their all Forms	ነረ(he/it was),አለ(it/he present)
	VPREP	Verb with non separated Preposition	በመጠ(after he came)
	VC	Verb with non separated Conjunction	በለኒ(as soon as he eat)
Adjective	ADJ	An adjective which is not attached with other categories	ፈየ(good),ጉመረ(white)
	ADJPREP	An adjective which is not separated from prepositions	ተጤም(with black)
Adverb	ADV	Tag for all types of adverb	ሁለግነ(always), አደደግነ(sometimes)
Preposition	PREP	preposition that are not attached with other word categories	በ(by),ተ(with),ለ(to)
Conjunction	C	Conjunctions that are not attached with other word categories	ዋ(and)ሀነግነ(or),ብዮንም(evenif)
Numeral	CN	Cardinal number	ሀድ(one),ሆሽት(two)...አስር(ten),1,2...
	ON	Ordinal number	ሀድለኜ(first)ሀምስትለኜ(fifth),1ለኝ፣
	CNPREP	Cardinal number with Preposition	በሆሽ(by two
	ONPREP	Ordinal number with Preposition	በሆሽትለኜ(in the second)
	ADJN	A numeral that function as an adjective	ሆሽት ላም(two cows)
Punctuation	PUN	Tags for all punctuations Marks	::, ፣,፤ ,: ,?, !
Interjection	INT	Interjection	ሀሹ!(wow),ኡጋሀ!(please)

learning algorithms.

1) Experiment 1: The first experiment is conducted to determine whether neural word embedding can encode syntactic and semantic information about words after adding Siltigna language raw text. The raw text was given to a Word2Vec tool with the dimension of one hundred and three hundred with window size of 1 and 2 to get word vectors of a given dimension. These word vectors were tested by finding ten closest words to some given words of different parts-of-speech.

Table III
MEASURING COSINE SIMILARITY BETWEEN W1 AND W2

Words	Dimension	Window size	Similarity
W1(በእሸረኜ) vs W2(እሸር)	100	1	0.019871779
W1(በእሸረኜ) vs W2(እሸር)	100	1	0.02333239
W1(በእሸረኜ) vs W2(እሸር)	300	2	0.091429584
W1(በእሸረኜ) vs W2(እሸር)	300	2	0.08025645

From the experiment result word vectors with the dimension of 300 with window size of 1 gives better value at the time of computing similarity between two words w1 and w2. These word vectors were tested by finding ten closest words to some given words of different parts-of-speech. See the following example top ten closest words for the word **"እሸር"**.

In order to assign the class of unseen word, by using the vectors of each word which is generated by word2vec tool computes similarity between the unseen word and the words that are found in the corpus with the given window size to predict its word class. After that the model compute the occurrence of that word (most nearest word to the unknown word) with specific tag and finally the model assign the same category of the word which found in the corpus based on which the probability of the word with specific class most frequently.

2) Experiment 2: The second experiment is done to test the effect of using pre-trained word embedding weights. To do this RNN model was built with three different embeddings such as arbitrarily initialized untrainable embedding, arbitrarily initialized trainable embeddings and trainable word2vec embeddings.RNN model was trained with different status of embedding weights. For conducting this experiment we uses a percentage split mechanism (80% for training and 20% for testing). Moreover, 20% of the training set was used to validate the model.The experiment was conducted using manually tagged Siltigna language corpus with different parameters and hyperparameters such as epoch of 300, Adam optimizer with learning rate of 0.001, loss function of categorical cross entropy and batch size of 128.

Experimental result under the second experiment:-

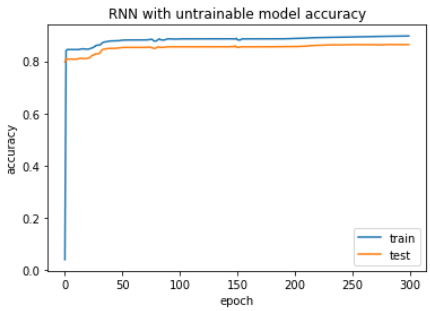

Figure 1. RNN with Untrainable Embedding

Experimental result of the second experiment is starting from arbitrarily initialized, untrainable embedding. In this experiment we won't use the pre-trained word embeddings. we use randomly initialized embeddings also we won't update the embeddings. Next we have done an experiment on RNN models with arbitrarily initialized, trainable embeddings(doing the change is the parameter trainable to true). Rest all remains the same as the above experiment on RNN model. Therefore all the parameters have become trainable means that trainable parameters are equal to the total number of parameters. Let's visualize the results of the experiment using second embedding type under the second experimental scenarios.

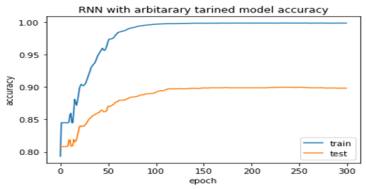

Figure 2. RNN with Arbitrary Trained Embedding

The accuracy of the model has increased significantly by 8.36% after training (fitting). By allowing the embedding weights to train, it increased from 89.03% to roughly 97.39%. As a result, embedding significantly affects how the network will function. Third experiment is weather word2vec embeddings improves our model or not. This experiment is done on RNN model with trainable word2vec embeddings. Recall that we had loaded the word2vec embeddings that we have pretrained in a matrix called 'embedding weights' and the network architecture is the same as the above two experiments. Let's visualize the results of the experiment using the third embedding type under the second experiment.

The accuracy, in this case, has gone even further to approximately 97.47%. The outcomes stepped forward significantly. This is because the model was already performing very well. Look at the summary of RNN model performance with different embedding weights of all the

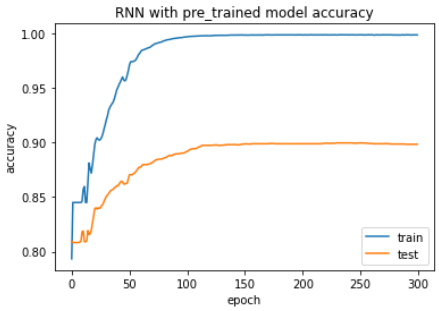

Figure 3. RNN with Pre-trained Embedding

three experiments under the second experiment on the following table.

The summary of the result shows that high performance is obtained by using pre trained embeddings. Based on this result we have concluded that using pre trained word embedding out performs better.

3) Experiment 3: The third experiment is conducted to check the performance of state of the art deep learning techniques on POS tagging for Siltigna language using neural word embedding as features. For conducting this experiment four deep learning RNN had been implemented by using neural word embedding as features and their performance is evaluated using accuracy as a metrics. To train each model the researcher uses the data sets which were prepared for training purpose with 300 and 128 epochs and batch size respectively with the networks of 64 cells and Adam optimizer with learning rate of 0.001, loss function of categorical cross entropy and trainable word2vec embeddings. Since the experimental result of the simple RNN model already discussed the experimental result under the second scenario with using pre trained embeddings.

When we compare the performance of the model with respect to the former one the accuracy result is improved significantly. Experimental result of GRU

Experimental result of Bidirectional-LSTM The same dataset was also tested on a bidirectional long short-term memory recurrent neural network (Bi-LSTM RNN) having four layers and 64 cells in each layer. The accuracy obtained is about 93.05% when we have compared this mode1 we have obtained better results among the others.

Therefore to sum up the third experiment let us see the experimental result of each algorithm as follows on Table V.

From table V as shown above standard recurrent neural networks having a total of three layers, 64 neurons on each layer, resulted in 92.49%accuracy. The other form of RNN what we call GRU with the same layer of RNN resulted with the accuracy of 92.55%, LSTM, which consists of 64 LSTM units in each layer, achieves an accuracy result with a score of 92.61%, Bi-LSTM, which has three layers and 128 smart neurons in each layer, was also checked on the

Table IV
Results for RNN with different embedding weights

RNN_model	Untrainable embedding	Arbitrary Trainable embedding	Pre-trained embedding
Accuracy	89.67%	92.13%	92.49%.
Deference	0	2.46	0.36
Loss	0.34	0.33	0.33

Table V
Summary of experimental results

Model	RNN	GRU	LSTM	Bi-LSTM
Training Accuracy	97.47%.	97.50%	97.66	97.73%.
Testing Accuracy	92.49%.	92.55	92.61	93.05

same datasets & achieved an accuracy score of 93.05%. Therefore, the researcher chooses the BiLSTM model from the experimental models mentioned above that were performed under the third experiment, because BiLSTM model out performs the other when we see the accuracy of each model. finally the researcher develops BiLSTM based Siltigna POS tagger model that has training accuracy of 97.73%, testing accuracy of 93.05% built with 300-epoch number and a 0.001 learning rate and percentage split of 80%:20%.

A. Conclusion and Future work

1) Conclusion: PoS tagging, which involves identifying and classifying words in a given text with their appropriate word categories, is one of the crucial applications of NLP. Different machine learning and deep learning approaches can be used to construct it. Each approach has its own advantages and drawbacks when we develop POS tagger. On this study, Siltigna PoS tagger is developed using different deep learning algorithms such as simple RNN, LSTM, GRU and Bi-directional LSTM. for this study sample corpus and tagset for the language are developed . For training to train the model, we have developed a Siltigna corpus of around 3500 sentences and 22 tagsets have been identified and dealt with in this research work. However, the tagset just indicates only word class without considering additional features like gender, number, tenses, etc. then from the entire corpus we have develop train and test sets. On this study 80% of total corpus has been used for training models and the remaining 20% have been also used for testing. To evaluate the effectiveness of the proposed method with neural word embeddings as features, different experiments were conducted on different deep learning algorithm. As a result, the accuracy obtained for simple RNN, LSTM, GRU and Bi-directional LSTM was 92.49%, 92.55%, 92.61% and 93.05% respectively. based on the result it is possible to conclude that Bi-directional LSTM outperformed all other three variants of RNN.

References

[1] G. Mamo and M. Meshesha, "Parts of Speech Tagging for Afaan Oromo," Int. J. Adv. Comput. Sci. Appl., vol. 1, no. 3, 2011.

[2] P. M. Nugues, "An Introduction to Language Processing with Perl and Prolog: An Outline of Theories, Implementation, and Application with SpecialConsideration of English, French, and German," Cogn. Technol., vol. 10, pp. 1–513, 2006.

[3] D. Kumawat and V. Jain, "POS Tagging Approaches: A Comparison," Int. J. Comput. Appl., vol. 118, no. 6, pp. 32–38, May 2015.

[4] T. Gebregzabiher, "Part of Speech Tagger for Tigrigna Language," 2010.

[5] A. Tukur and A. Sa, "Parts-of-Speech Tagging of Hausa-Based Texts Using Hidden Markov Model," no. July, 2020.

[6] "Liddy, E. D., (2003). Natural Language Processing. In Encyclopedia of Library and Information Science, 2nd Ed. Marcel Decker.

[7] S. Asemie, T. Bekele, and Z. Solomon, "A Hidden Markov Model-based Part of Speech Tagger for Shekki ' noono Language," no. February 2022, 2019

[8] J. Singh, N. Joshi, and I. Mathur, "P ART O F S PEECH T AGGING OF M ARATHI T EXT," vol. 3, no. 2, pp. 35–41, 2013

[9] M. Argaw, "Amharic Parts-of-Speech Tagger using Neural Word Embeddings as Features Amharic Parts-of-Speech Tagger using Neural Word Embeddings as Features," 2019.

[10] W. B. Demilie, "Parts of Speech Tagger for Awngi Language," 2019, [Online]. Available

[11] G. M. Wegari, "Parts of Speech Tagging for Afaan Oromo,"

[12] B. F. Shirko, "Part of Speech Tagging for Wolaita Language using Transformation Based Learning (TBL) Approach," vol. 10, no. 9, [Online]. Available:

[13] W. B. Demilie, "Analysis of implemented part of speech tagger approaches: The case of Ethiopian languages," Indian J. Sci. Technol., vol. 13, no. 48,

[14] Shereen Khoja. APT: Arabic Part-of-speech Tagger. Computing Department, Lancaster University Lancaster LA1 4YR, UK.

[15] Eric Brill. 'A Simple Rule-Based Part-of-Speech Tagger'. In: proceeding of the third conference on Applied Natural Language Processing, Trento, pp. 152-155 .

[16] Tunga Gungor, A composite approach for part of speech tagging in Turkish, Bogazici University, Istanbul, Turkey .

[17] P. Wang, Y. Qian, F. Soong, L. He, H. Z. preprint arXiv, and undefined 2015, "Part-of-speech tagging with bidirectional long short-term memory recurrent neural network.

[18] M. Getachew, "Automatic Part of Speech Tagging For, Amharic Language an Experiment Using Stochastic Hidden Markov (Hmm) Approach," 2001.

[19] Z. Mekuria, "Design and Development of Part-of-speech Tagger for Kafi-noonoo Language," 2013.

[20] S. Mohammed, "Using machine learning to build POS tagger for under-resourced language: the case of Somali," Int. J. Inf. Technol. 2020 123, vol. 12, no. 3, pp. 717–729, Jun. 2020.

[21] S. Hirpssa and G. S. Lehal, "POS Tagging for Amharic Text, A Machine Learning Approach,2001.

[22] I. Gashaw, "Machine Learning Approaches for Amharic Parts-of-speech Tagging," no. December, pp. 69–74, 2018.

[23] Fitsum Gizachew, "DEVELOPING PART OF SPEECH TAGGER FOR GURAGIGNA LANGUAGE," Mar. 20, 2020.

[24] K. Desta, "Part of Speech Tagger for Hadiyyisa Language," 2019.

[25] W. Leslau, "Eeva H. M. Gutt - Hussein Mohammad: Silt'e - Amharic - English Dictionary (with Concise Grammar by Ernst-August Gutt," Aethiopica, vol. 3, no. 1, pp. 251–254, Sep. 2013.