Part of Speech Tagger Using Empirical Evaluation of Neural Word Embedding and N-Gram Approaches for Koorete

Agegnehu Ashenafi Faga (Corresponding author)
Department of Computer Science, Wachemo University, Hossana, Ethiopia
Email: agegnehua091@gmail.com

Tesfaye Bayu Bati
Department of Computer Science, Hawassa University, Hawassa, Ethiopia
Email: tesfayebayu@hu.edu.et

Wubetu Barud Demilie
Department of Computer Science, Wachemo University, Hossana, Ethiopia
Email:wubetubarud@gmail.com

Melese Zekiwos Segaro
School of Computing, University of Eastern Finland, Joensuu, Finland
Email: msegaro@uef.fi

Abstract

The Ethiopian low resourced local language - Koorete is spoken by the Koore Zone people, who are located in the southern part of Ethiopia. The language is also used as the medium of instructions beyond 350 thousand Koore nations and some other people beyond its border. The language follows the sentence structure of **"**Subject (Zeere utaade) + Object (efaxe) + Verb (Hanta beyiisaxe)**".** This paper aims to develop Part of Speech (POS) tagger using the empirical evaluation of Neural Word Embedding and N-gram-based approaches for Koorete. According to this scope, neural word embedding represents the sequence of labeling and distribution of words into vectors (Word2Vec) by applying the bidirectional long short-term (Bi-LSTM) recurrent neural network (RNN) memory model. This model also achieved state-of-the-art POS tagging prediction accuracy by comparing it with the classic N-gram frequency prediction since it is triggered in the research question. Demonstration is made to the Bi-LSTM RNN and N-gram model on the same Koorete POS tagger (KPT) manually annotated corpus. This KPT corpus used 1718 sentences with about 33,220 words and divided the corpus into 90% for training and 10% for testing. Finally, the Bi-LSTM RNN word embedding POS tagging approach performed 98.53% for training and 98.49% for testing; whereas, the N-gram POS tagging approach performed about 97.10% for training and 77.29% for testing. These results could lead to the conclusion of Bi-LSTM RNN model performed better than the N-gram model accuracy.

Keywords: POS Tagging, Word2Vec representation, neural word embedding, N-gram, Deep Learning, Koorete languages

DOI: 10.7176/CEIS/17-1-03

Publication date: February 28th, 2026

1. Introduction

These days, people are highly interacting with each other within a fraction of the time as the world is connected to one village through the Internet. The interaction needs to know how to attractively communicate with its customers in a formal, coordinated way. This means humans use the natural language of their mother tongue, and second, or third languages to express their ideas, culture, and feelings with good communication. So, the way of expressing a natural language to represent an idea is termed natural language processing (NLP). NLP could be represented at different levels, such as word level, phrase level, sentence level, or semantic level. Language processing needs improvement in documentation status for information, per the advancing technology like Bi-LSTM neural embedding [**1**]. Knowing one's language processing starts from knowing its part of speech; which word class the given word needs to be labeled/tagged; its syntactic and semantic values, and the phrase and sentence contexts[**2**][**8**]. Computers cannot easily understand human language syntax and its semantics in a sentence before automation. It is becoming difficult for humans to analyze and get the necessary computerized content because the data requirements of each natural language are increasing. Therefore, we need to develop a computer application to adapt a machine language concerning natural language, so that a computer can understand natural language as humans do. So, the Koorete has to be developed in advanced technology aspects.

For its share of contribution, this proposed study is developing Koorete part of speech (POS) tagging using neural word-level embedding and N-gram N-gram-based approach with a corpus named Koorete POS Tagger corpus.

The Koorete language is spoken by the Koore Zone people in South Ethiopia. The language is also used as the medium of instructions beyond 350 thousand Koore nations and some other people beyond its border. It belongs to the family of North Omotic languages. Koorete is written using the Latin script (or 'Diizo Beyta' in the Koorete language). The language has 31 consonants (Artaxita), 5 vowels (Arxaxita), and one more symbol called pooxhe ('). The language follows the sentence structure of "Subject (Zeere utaade) + Object (efaxe) + Verb (Hanta beyiisaxe)".

Due to the existence of different natural languages, there are different sentence structures, and one's language morphological features are relatively different from one another. By itself, any natural language is not easy to process as a language is ambiguous, subjective, and complex [**20**]. Because of these, many researchers have used different learning approaches to automate, such as rule-based [**36**; **7**], probabilistic or corpora-based [**22**; **8**], neural network [**2**; **3**], and deep learning approaches [**1**; **6**]. For instance, this research used the Bi-LSTM RNN deep neural network and N-gram statistical approach for its realization.

The reason why this study used Bi-LSTM RNN is that this approach gives the state-of-the-art performance results of about 97% and with tagging accuracy in the new corpus [**2, 3, 5, 6**]. This literature discusses that Bi-LSTM RNN is a 'state-of-the-art' algorithm for better accuracy. This means it is the current and recent trend that most researchers are using this approach for POS tagging on several languages without the consideration of morphological features. Besides, it is a deep learning algorithm. This approach operates well on a dataset minimum size of **10K**. This study used an N-gram statistical-based learning approach because most of the Ethiopian language researchers are using statistical-based learning approaches. Because the Ethiopian languages need new corpus preparation for statistically based learning patterns, and most of the time, they do not need rule-based approaches. Because of this, Ethiopian languages do not have computerized pre-prepared rules, lexicons, dictionaries, and phonemes. For the NLP applications, especially during POS tagging, mostly Ethiopian researchers use HMM and N-gram to predict word class. Besides, according to Grigori Sidorov et al. [**37**], syntactic N-gram-based techniques are predominant in modern NLP and its applications. This research paper has contributed to the following tasks localized for the POS tagger to the Koorete language, which is one of the Ethiopian languages.

- It developed and provided Koorete POS Tagger annotation corpus
- It adapted appropriate tagsets to both models and then evaluated the empirical measurement between the Bi-LSTM RNN model and the N-gram language model approaches
- It concluded that the better performance of the Bi-LSTM RNN model over the N-gram language model

The rest of the paper is organized into different but interrelated subsections. The paper begins by discussing the literature review of related works in Section 2, the proposed research methodology, the experimental analysis and its result discussion in Section 3, and the conclusions and future enhancement in Section 4.

1.1. Statement of the Problem

As POS tagging is a precondition for most NLP applications, it uses several sources of information such as dictionaries, lexicons, rules, and so on. It means a word may belong to more than one category in a dictionary. Named entity recognition, word sense disambiguation, and syntactic parsing are some of the advanced high-level NLP applications [**4**]. When developing high-level NLP applications, POS tagging is a basic task to be practically, theoretically, and feasibly applied to real-world communication extension. For example, in Word sense disambiguation (WSD), WSD does the task of understanding the correct sense or meaning of a word in a given context; whereas knowing the POS tag in WSD helps in ordering sentence structure and proposing the next word.

Koorete needs to have a researcher's contribution to POS tagging in high-level NLP applications to enhance language resources. Up to the extent of my exploration, no studies have been done on Koorete language POS tagging [**1**]. This shows that the Koorete language has no good documentation status, meaning it is not rich in language resources, which need to be applied in advanced NLP applications and linguistic study [**1**]. Therefore, it is under-resourced. Koorete is being delivered as a school subject for Elementary students. This may be a crucial new research area for some NGOs to intervene [**1**]. Despite this fact, there is the morphology of Koorete conducted especially on Verb morphology (Beletu, 2003). Also, I have interviewed a Kooreete language expert from Dilla Teachers College of Education, where there are many Kooreete language instructors; but there has not been any Kooreete POS tagging contribution yet.

Koorete POS tagging also needs to reach the state-of-the-art tagging in the performance of the other languages' POS tagging mode[2]. For this reason, this study uses a neural word embedding model. This model takes into action a Bi-LSTM RNN in sequence labeling to predict a tagging probability distribution of each word. It achieves performance to the extent of prediction in language modeling, language understanding, and machine translation without using morphological features [2][3]. The other problem is the absence of a prepared Corpus in the Koorete language for POS tagging. For POS tagging, any language has to have its language corpus as a dataset. Corpus-based POS taggers build models from the training dataset using one or more algorithms and apply the models to unseen instances of the language [8].

1.2. Research Questions (RQ)

The following are raised in this study to implement their prototype with a convenient and enough dataset, are here as follows.

RQ1: How could the Bi-directional LSTM RNN-based neural word embedding POS tagging approach perform concerning the N-Gram Statistical POS tagging approach for the Koorete language?

[*How could neural word embedding simplify Koorete POS tagging relative to the N-gram-based statistical approach?***]**

RQ2: What are the most appropriate tagsets that need to be used for Koorete POS tagging?

[*Adapt appropriate tagsets***]**

2. Literature Review of Related Works

In this study, in Table 1 below, the author has analyzed the literature on various POS taggers and their word class annotations, as well as the models/techniques that have been used, illustrated.

Table 1. Literature Review Summary

Source and Year	Objective	Methods	Dataset	Performance
Sintayehu Hirpassa et al.[1], 2023	Improving POS tagging with DNN	BiLSTM-CRF	Germen and English, 321 K words	97.85%
Anastasyev et al. [9], 2018	Improve POS tagging performance using Character embedding on English and Russian languages	Bi-LSTM, RNN, CharFF, Word2vec	PTB(45 tags), Gikrya(304 values), Syntarus (908 values)	Accuracy achieved between 96% and 98.5%
S.Ezhilarasi et al. [2], 2021	POS tagging classification and predicting	Bi-LSTM, RNN	Tamil language	88.88% accuracy
Pinky et al. [7], 2019	Maximize human interpretability by semantic similarity between texts	Word2vec with Cosine similarity	2000 news articles of which D1-D5 documents comparison	Accuracy of 76.8%, 75.9%, 76.05%
Birhanesh F.[22], 2020	Predict next words word category	TBL	26 tag sets 1134 sentences 14358 words	92.96% accuracy on Trigram tagger
N.X. Bacha et al. [24], 2018	Specify word class tag	CRF	4000 sentences from Facebook	88.26% & 88.92% rated
J.H. Yousi et al. [25], 2019	Predict probability of tagging	HMM	40 sentences 458 words	97.6% and 97.4%
T.B. Shahi et al. [26], 2013	Handle challenging features in binary forms	SVM	10775 tokens	93.27% for known and unknown words

Grigori Sidorov et al. [37], 2022	Training machine learning SN-grams as features	Syntactic N-grams, SVM	Profile size of 400 for bigram, and 700 for trigram	SVM and SN-gram achieved 100% both with bigram and trigram
Erick R. et al. [5], 2015	How map unlabeled data to vector space	MEMM	Mac-Morpho corpus 452 sentences	93.1% accuracy
Bin et al. [6], 2019	Evaluate word embedding	DNN, N-gram, Dictionary	26 tag sets 1100 sentences	92% accuracy
Serkan et al.[21], 2018	SMS classification into spam and ham	Word2vec with semantic relations	5574 lines of short messages 4827 ham and 747 spam	Accuracy rate of 99.64%
Khwlah et al.[20], 2019	Automate NLP applications	LSTM RNN and word2vec	77,915 words	99.55% for tagging morphemes and 97.33% for tagging words
Duressa Tamirat [32], 2016	Corpus-based word prediction	N-gram tagger	set of 2,242 sentences with 37,272 token words	83.84%, 61.4%, 54.8% of unigram, bigram and trigram respectively

Notable: Based on Table 1 above, literature summary and its approaches used for POS tagging performance simplicity, and NLP applications used to represent one's language on the computer for the sake of computerized resource building, this study also has cascaded appropriate approaches. So this research studies the empirical evaluation of neural word embedding (Bi-LSTM RNN) and N-gram-based statistical approach on the KPT corpus of 33,220 words. It means these approaches assign POS tags to words based on context similarity, and then this study distributes words in the vector space.

Firstly, the reason that the Bi-LSTM RNN is used could be because it gives state-of-the-art performance results of about 97% and with tagging accuracy in the new corpus [2, 3, 5, 6]. This literature says Bi-LSTM RNN is a 'state-of-the-art' algorithm for better accuracy. This means, it is the current and recent trend that most researchers are using this approach for POS tagging on several languages without the consideration of morphological features. Besides, it is a deep learning algorithm. This approach operates well on a dataset of a minimum size of **10K**. ***Secondly***, the reason why this study uses an N-gram statistical-based learning approach is that most of the Ethiopian language researchers are using statistical-based learning approaches. Besides, the Ethiopian languages need new corpus preparation for statistical-based learning patterns. Most of the time, they do not need rule-based approaches. Because the Ethiopian languages do not have computerized pre-prepared rules, lexicons, dictionaries, and phonemes. In NLP applications, especially for POS tagging, most Ethiopian researchers use HMM and N-gram to predict word class. Besides, according to Grigori Sidorov et al. [37], syntactic N-gram-based techniques are predominant in modern NLP and its applications.

The language, Koorete, is not well studied in linguistics, computational, or computer-aided NLP applications. This is why choosing POS tagging approaches on N-gram statistical and neural word embedding by many recent researchers in contributing their shares could be the choice.

2.1. Why was Deep Neural Word Embedding Chosen?

Most works use words as the smallest units in the compositional architecture, often using pre-trained word embedding [3]. Word embedding is one of the most popular representations of document vocabulary |V| to word vectors (Word2Vec) as inputs. It is the unified name for a set of language modeling and feature learning techniques in NLP, where words or phrases from the vocabulary are mapped to vectors of real numbers. It transforms human language meaningfully into a numerical form. This means word vectors are simply arrays of numbers that represent the meaning of a word concerning its index position given in the vocabulary. Word2Vec is a dense two-layer neural network representation. Why Word2vec? Because it (1) preserves relationships between words and their index position, (2) gives better results in lots of deep learning applications, and (3) deals with the addition of new words to the vocabulary. This embedding is simply a process of word embedding for a current word prediction based on the given surrounding words. It is a distributed representation of a word that allows deep learning methods to perform well on challenging NLP problems. This embedding enables words to have similar meanings in a similar context and could be represented similarly by putting them in the same vector space. This is one of the basic advantages of the reduction of out-of-vocabulary impact [8]. This is possible because words will not be completely unknown as long as they have feature vectors, even if they may not be seen in the training dataset. Because of the use of neural networks, this embedding method generates the Word2Vec mappings since a word is the underlying input representation. This brings the idea of generating distributed representations. This is possible because words will not be completely unknown as long as they have vector features, even if they may not be seen in the training dataset. Because of the use of neural networks, this embedding method generates the Word2Vec mappings for raw words since a word is the underlying input representation.

The term '*deep'* refers to the number of problems learning from N-to-N of hidden layers in the neural network as of the N >= 2. Deep learning is a class of machine learning techniques that performs much better on unstructured data. It is outperforming current machine learning and tackles problems on which shallow architectures (e.g., word2vec) are affected by the inflexibility of dimensionality [14]. It enables computational models to learn features and layered model inputs progressively from data at multiple-level abstractions [13]. Deep neural networks are successful in Supervised learning, Unsupervised learning, Reinforcement learning, hierarchical learning, as well as hybrid learning. The algorithms in deep learning use the back-propagation algorithm that discovers complex structures from large datasets by looking for the previous and next-word context. Back-propagation solves an optimization problem using a gradient-based calculation method for each iteration in the Bi-LSTM RNN model [13], and so Bi_LSTM is sketched below in Figure 1.

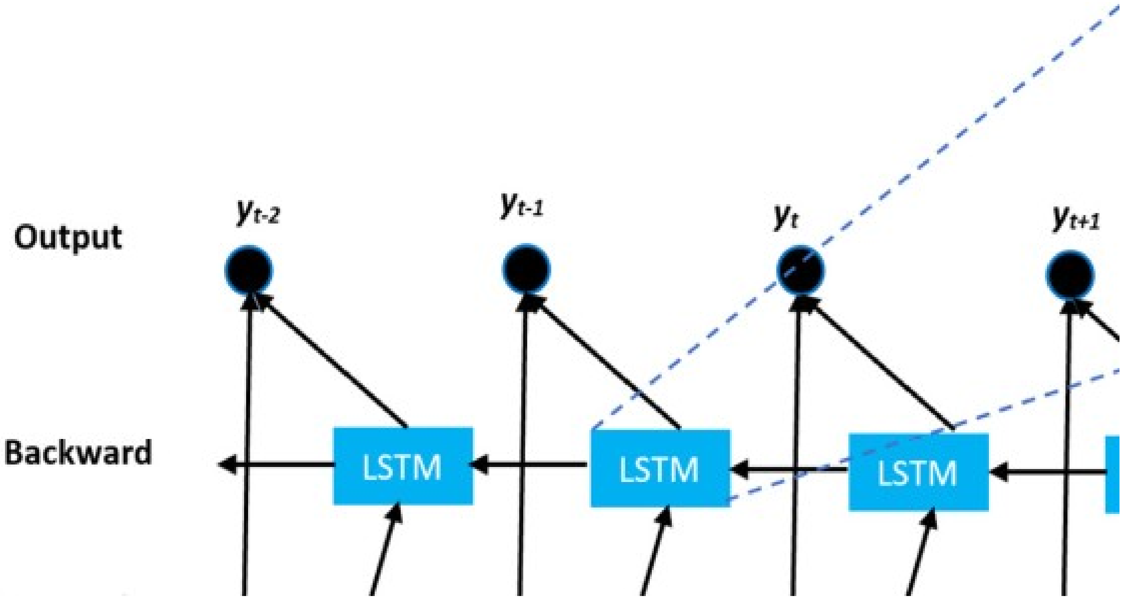


Figure 1. Overview of the Bidirectional LSTM model (picture credit: [35])

Deep learning architectures include recurrent neural networks (RNNs), convolutional neural networks (CNNs), and deep neural networks (DNNs) that are used in different areas, including NLP applications. From the possible architectures of RNNs, their extension, LSTM RNNs, is used in this study. The better performance of LSTM is about 97.40% accuracy, which was observed when Peilu Wang et al. applied English with word embedding as a feature [2]. The architectural approach of this study is tested on the Koorete POS tagger corpus. Why deep learning is needed can be answered as it gains supremacy in terms of accuracy when trained with large amounts of data as these days in the era of big data.

2.2. Why were Regular Expression and N-gram Approaches Chosen?

This proposed study uses the N-gram language model up to the Trigram tagger. This study used the chain rule for the N-gram-based statistical POS tagging approach to estimate each Trigram probability P(w) from the tagged corpus. This chain rule is illustrated as following herein below.

$$P(w_{i-2}|w_{i-2},w_{i-1}) = C(w_{i-2},w_{i-1},w_i) / C(w_{i-2},w_{i-1}) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ (1)$$

where C refers to the chain rule conditional probability of the sequence of words in a sentence

This chain rule puts the conditional probabilities for each N-gram tagger from the lower tagger (default and regular expression) up to the Trigram tagger. These are the comparison of the target word to its previous and next words such as the *unigram tagger* evaluates the probability of a target word to its one previous and next word, and the *bigram tagger* evaluates the probability of a target word in comparison with its two previous and next words, the *trigram tagger* evaluates the probability of a target word in comparison with its three previous and next words. These are evaluated as the following formulae.

$$\text{Trigram model: } P(w_i|w_1, w_{2,\ldots}w_{i-1}) \approx P(w_i\,|w_{i-2,}\, w_{i-1}\,) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ (2)$$
$$\text{Bigram model: } P(w_i|w_1, w_{2,\ldots}w_{i-1}) \approx P(w_i\,|w_{i-1}\,) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ (3)$$
$$\text{Unigram model: } P(w_i|w_1, w_{2,\ldots}w_{i-1}) \approx P(w_i\,|w_i\,) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ (4)$$

For example, the probability of Koorete language sentence “*Zeere denxoy accaaka axe miidheko*” using a Trigram tagger could be:

P(Zeere denxoy accaaka axe miidheko) = P(miidheko|Zeere denxoy accaaka axe)
= P(miidheko|accaaka axe)

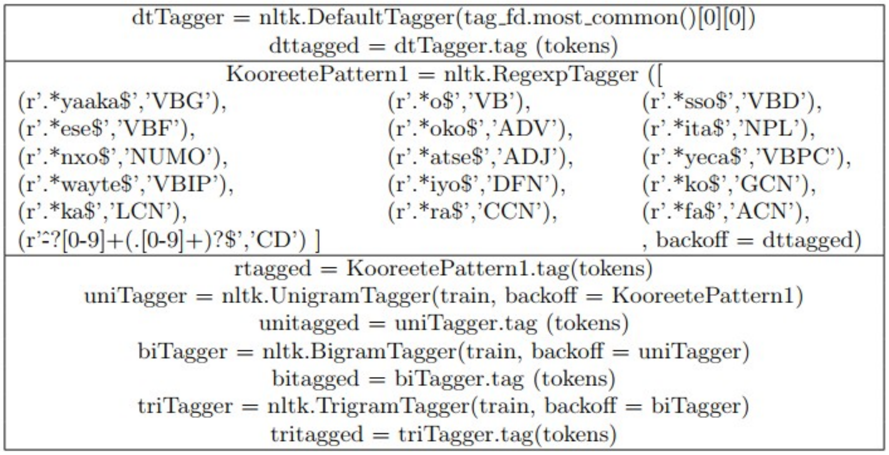
2. Combining Taggers using the ‘Backoff’ Keyword

Backoff is used to handle the probability of POS prediction down to the lower tagger when the higher tagger fails to compute and determine POS tags [2]. It addresses the increase in accuracy, and its coverage is to the more accurate algorithms calling back to the lower tagger. For this, we combine the results of a trigram tagger, a bigram tagger, a unigram tagger, a Regexp tagger, and a default tagger using the backoff keyword, as follows:

1. Try tagging the token with the higher-gram tagger but in this Trigram tagger.
2. If the Trigram tagger cannot find a tag for the token, try the Bigram tagger.
3. If the Bigram tagger cannot find a tag for the token, try the Unigram tagger.
4. If the Unigram tagger cannot find a tag for the token, try the Regexp tagger.
5. If the Regexp tagger cannot find a tag, use a Default tagger.

For example, the code segment has used ‘**train**’ as the training sentences; ‘**tokens**’ as the validation checkup of the new corpus; ‘**tag_fd**’ parameter in the default tagger to refer to the frequency distribution of tags in the training data. As the following, Table 2 discusses the rule-based language model incorporated in the N-gram taggers and regular expressions for combining with the backoff keyword to find tags easily and fulfill the principle of the higher N-gram to the lower level N-gram taggers. So, Table 2 regular expression inclusion into N_gram taggers is taken from the KPT corpus prepared as its dataset tag sets are mentioned in Table 3 below.

Table 2. Combining N-gram with the ‘backoff’ Keyword



```
                 dtTagger = nltk.DefaultTagger(tag_fd.most_common()[0][0])
                           dttagged = dtTagger.tag (tokens)
                       KooreetePattern1 = nltk.RegexpTagger ([
(r’.*yaaka$’,’VBG’),              (r’.*o$’,’VB’),          (r’.*sso$’,’VBD’),
(r’.*ese$’,’VBF’),                (r’.*oko$’,’ADV’),       (r’.*ita$’,’NPL’),
(r’.*nxo$’,’NUMO’),               (r’.*atse$’,’ADJ’),      (r’.*yeca$’,’VBPC’),
(r’.*wayte$’,’VBIP’),             (r’.*iyo$’,’DFN’),       (r’.*ko$’,’GCN’),
(r’.*ka$’,’LCN’),                 (r’.*ra$’,’CCN’),        (r’.*fa$’,’ACN’),
(r’^-?[0-9]+(.[0-9]+)?$’,’CD’) ]                           , backoff = dttagged)
                      rtagged = KooreetePattern1.tag(tokens)
          uniTagger = nltk.UnigramTagger(train, backoff = KooreetePattern1)
                        unitagged = uniTagger.tag (tokens)
              biTagger = nltk.BigramTagger(train, backoff = uniTagger)
                         bitagged = biTagger.tag (tokens)
              triTagger = nltk.TrigramTagger(train, backoff = biTagger)
                         tritagged = triTagger.tag(tokens)
```

The discussions realize that tag sets such as VB, VBP, VBG, VBF, ACN, CCN, LCN, CD, …, etc. are presented in Table 2 above in the Koorete POS Tagger (KPT) corpus. The default tagger used in this study is noun NN, obtained by counting the most frequent tags distribution using the default tagger from the corpus. Besides, the regular expression patterns are shown in this section with the hand-coded pattern using the Regexp tagger.

3. Experimentation on Research Methodology

3.1. Proposed Framework

The neural feature extractor is composed of an input layer, a hidden layer, and an output layer [2, 16], and it is responsible for creating dense vector representations for each word feature [16]. The Input Layer takes the vector representations of each word as input in the corresponding word features. Each word feature is a sequence of tokens, represented as vectors. The vector representation for each token is composed of the word embedding corresponding to the token, concatenated with a one-hot representation of the token’s POS tag [2, 6]. The hidden layer takes the token vectors as input and processes them in a forward and a backward pass. In the forward pass, the content value of the hidden layer at a specific time step is calculated using the value of the input at the current time step, and the content value of the hidden layer in the previous time step at the backward. The output layer is a function that automatically assigns weights to the output of the hidden layer at each time step and calculates the weighted sum of the outputs using their corresponding weights using a softmax function.

Here is the flow diagram of the proposed approach framework, in which the input from the corpus is, fed to the Word2Vec skip-gram model and Bi-LSTM RNN model besides to the N-gram tagger. So, this study compares the relative performance of the neural network, and the stochastic N-gram tagger approach is shown in Figure 2 below.

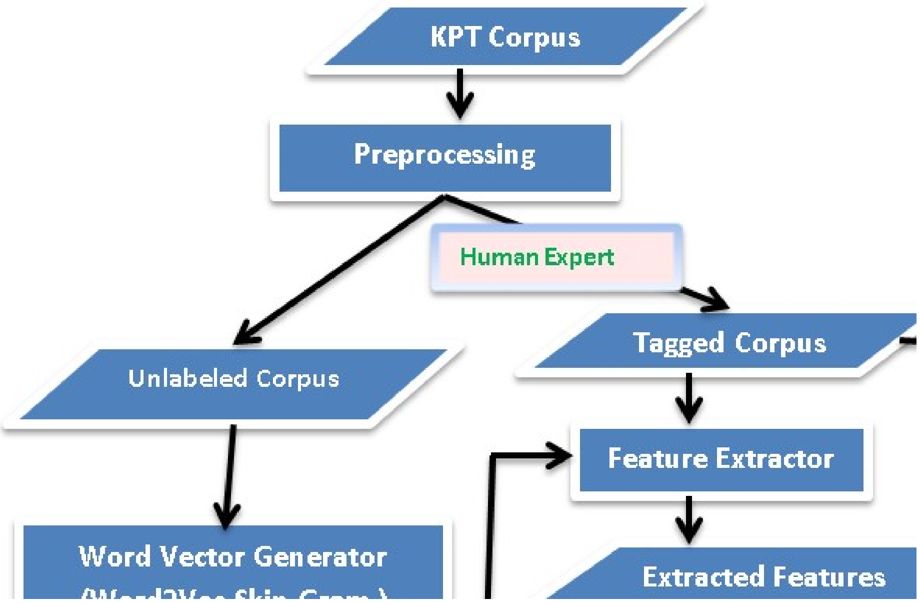


Figure 2. Proposed Approach Framework

The *word vector generator* generates the complimentary real number for each word; the f*eature extractor* processes the task of congruent words to real numbers extracted feature; the *model building* performs the task of training and testing the model.

3.2. Proposed Language Corpus Development and its Tagsets

The corpus is collected from A-Z Koorete language dictionary, New Testament Bible of Koorete, Koore Zone former named Amaro Special Wereda Culture and Tourism Office, Book of '**Dicchoo'** published in 2006 E.C[38], School Dictionary [39] , Koore Zone former named Amaro Special Wereda Education Office, and Koorete language department at Dilla College of Teacher Education. The Koorete dictionary has embedded tags of a Koorete part of speech. Most portion of the Koorete POS Tagger corpus is prepared from two sources. The first source is Matthew Gospel of New Testament Bible of Koorete, and the second one is Book of 'Dicchoo'. Book of 'Dicchoo' is translated from Amharic to Koorete by Tizita Fabric, who was a graduate student of Koorete language at Dilla College of Teacher Education in 2012 E.C. This book narrates about Koore cultures, norms, stories, heritages, and jewelries, municipality and Kebeles structure, Koore Kings administration hierarchy, cultural food preparation, and staple food of Koore termed 'Inset' (or '**Shuncha'** in Koorete). The other source is Koore Kings funeral system including its material which is house-like shaped called '**Komboxe'**, wailing ceremony, weeping orchestra during burial, burial style, and its condolences. Besides, the other source is the marriage process from dowry to wedding in Koore people culture. Finally, the corpus is presented to Dr. Samuel Zinabu, who is the Koorete language expert. He had delivered Koorete language at Hawassa College of Teacher Education for Koorete department students until the Koorete department shifted to the Dilla Teachers Training Center (TTC), Ethiopia. The size of corpus is prepared, which has **1718** sentences for ***33220*** words. This large size is why the corpus of Bi-LSTM RNN developed above needs a minimum size of **10k** for enough data usage, and for efficient performance on the skip-gram algorithm.

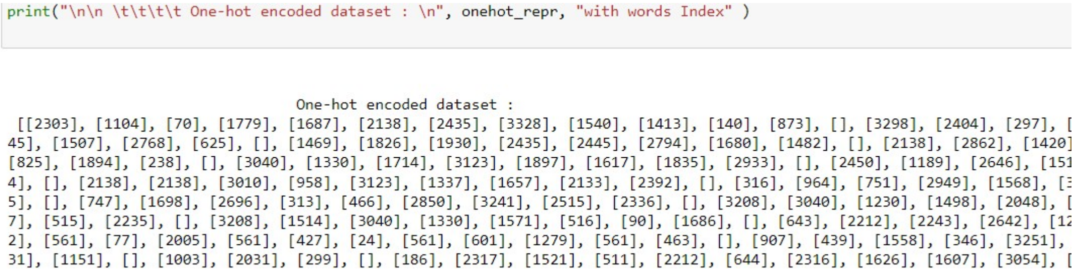
This corpus preparation is written using the Koorete language in Latin script, and its annotation is done with the help of the language experts at the College of Teachers' Education at Hawassa and Dilla, Ethiopia. According to the preparation of this corpus, the number of tag sets consisted of **11** original tag sets and expanded to **24** tagsets. There were not any other tag sets prepared yet in a documented and computerized fashion. So, the following Table 3 of KPT tag sets is cascaded from the analysis of Koorete Segmental phonology [17], Part of Speech Tagging for the Wolaita Language using the Transformation Based Learning (TBL) Approach [22], and Morphology of Koorete [28] for relativity representation. This is because the Wolaita language has relatively similar phonetics in addition to the languages written in Latin alphabets. The proposed Koorete KPT tag sets are shown in Table 3 below.

Table 3. The KPT Tag set having **11**-original and expanded **24**-tagsets

No	Basic Tags	Derived Tags	Definition	Example
1	Noun	NN	Noun	Adey **accho**/NN muudo (Father ate a meat).
		NPL	Plural Noun	Gerri meqet**ita**/NPL uqusuwa (You can approach intimacy with genealogical descendence).
		CAN	Ablative Case	Guliixhia**fa**/ACN zayte nu degesso.
		LCN	Locative Case	Buushe qaam‘o utulma**ka**/LCN gusuwa.
		CCN	Commutative Case	Ade wonto**ra**/CCN kardiitofu.
2	Pronoun	PRPRON	Proper Pronoun	**Hawassa**/PRPRON katamay/PRPRON orijheena akooko (The town of Hawassa is very vast).
		DSPRON	Distributive Pronoun	Neera **Wola**/DSPRON handzo gaddhesa hanguwaso (Today I wil not go to the market).
		DMPRON	Demonstrative Pronoun	**Yede**/DMPRON kanay woxhenike beewa(Look how that dog is running).
		IRPRON	Interrogative Pronoun	Doo **ayiidewu**/INTPRON ne yoodo (When did you come)?
3	Adjective	ADJ	Adjective	**Jhileta**/ADJ maatawo wonguwa (Buy green grass only).
4	Adverb	ADV	Adverb	Ne busshi **iitanako**/ADV modhe (Your girl is very beautiful).
5	Verb	VB	Verb Simple	Ne busshi iitanako **modhe**/VB (Your girl is very beautiful).
		VBP	Verb Phrase	Ne bushanchey iitanako modho**sso|oose**/VBP (Your girls are all very beautiful).
		VBG	Verb Gerund	Zine hano wotiyako nu **yaaca**/VBG (Yesterday this time, we were digging)
		VBF	Verb Future	Haya zawa hanta**ko**/VBF ta han**te**
6	Conjunction	SCONJ	Subordinate Conjunction	Yede **badena**/SCONJ e’uwa (stand aside to the below position).
		CCONJ	Correlative Conjunction	Muwa **ooyne**/CCONJ denxuwa (Eat or else take it off).
7	Numerical	NUMO	Ordinal Number	Handzoy **lanxo**/NUMO wontako (Today is 4th day).
		NUMC	Cardinal Number	Handzo **lam’i**/NUMC keexita nu keexho (Today we made two houses).
		CD	Cardinal Decimal	36.96
8	Preposition or Postposition	PREP or POST	Preposition or Postposition	Horobhilay yeke saha **e**/PREP hodho (Airplane landed to the land).
9	Punctuation	PUNC	Punctuation	Se axi modheko**.**/PUNC
10	Interjection	IJ	Interjection	**Eyyi**/IJ aba iita axewu!
11	None	NONE	None	Used for words having no tags in the corpus

3.3. Word Embedding using Indexing and One-Hot Representation

Applying the one-hot vector representation is very cumbersome because of its rigid vocabulary |V| size in the limited vector space. There may be also repeated usage of the same words many times in the vocabulary in a one-hot encoding indexing. Besides, at the time of trying to add new words that are out of indexed to the vocabulary, it is not possible to train and test by embedding in the middle of the vocabulary of already given its indexes[5] [6]. The indexes already given are not being changed by any means other than giving a new index for the whole vocabulary |V'|. Therefore, instead, it is preferable to embed the word with merely the unique words indexing in the vocabulary rather than applying indexes to the same word being used many times in the vocabulary. This rigid and cumbersome way of representation might be fixed with word embedding method of giving only one array index for a word throughout the Vocabulary even words is used repeatedly. One-hot representation removes word redundancy in the vocabulary as it is represented in Figure 3 below with words indexing on behalf of one-hot.

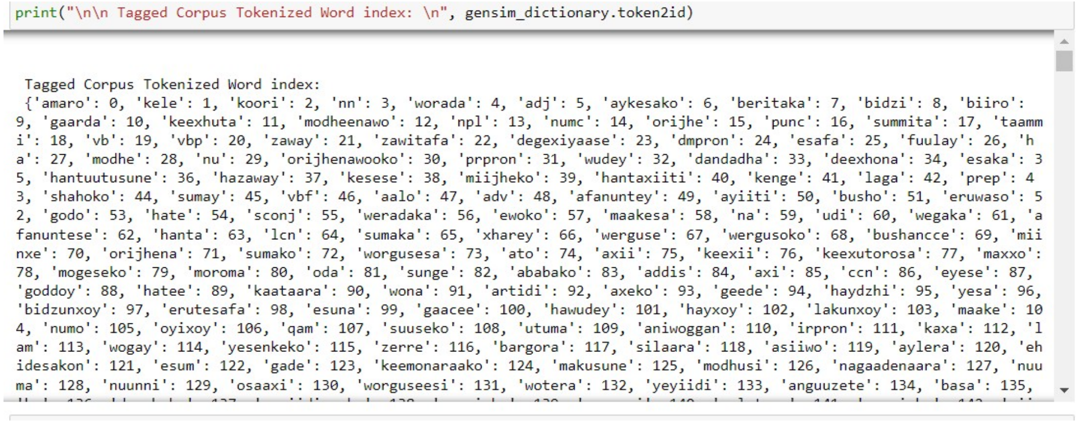


```
print("\n\n \t\t\t\t One-hot encoded dataset : \n", onehot_repr, "with words Index" )
```

```
                              One-hot encoded dataset :
 [[2303], [1104], [70], [1779], [1687], [2138], [2435], [3328], [1540], [1413], [140], [873], [], [3298], [2404], [297], [
45], [1507], [2768], [625], [], [1469], [1826], [1930], [2435], [2445], [2794], [1680], [1482], [], [2138], [2862], [1420]
[825], [1894], [238], [], [3040], [1330], [1714], [3123], [1897], [1617], [1835], [2933], [], [2450], [1189], [2646], [151
4], [], [2138], [2138], [3010], [958], [3123], [1337], [1657], [2133], [2392], [], [316], [964], [751], [2949], [1568], [
5], [], [747], [1698], [2696], [313], [466], [2850], [3241], [2515], [2336], [], [3208], [3040], [1230], [1498], [2048], [
7], [515], [2235], [], [3208], [1514], [3040], [1330], [1571], [516], [90], [1686], [], [643], [2212], [2243], [2642], [12
2], [561], [77], [2005], [561], [427], [24], [561], [601], [1279], [561], [463], [], [907], [439], [1558], [346], [3251],
31], [1151], [], [1003], [2031], [299], [], [186], [2317], [1521], [511], [2212], [644], [2316], [1626], [1607], [3054], [
```

Figure 3. One-Hot with Word2Vec Indexed Representation

When representing a word as per its index position in the vocabulary, it will be given *one vector*, and all others index position *zero* [5] [6]. The main idea here is that every word can be converted to a set of real numbers of N-dimensional vector |V| for good representation. This word-to-vector representation could capture the context of a word in a document, semantically, and syntactically similarity, relative to other words, etc. The basic idea of word embedding could be words of similar contexts have similar meanings by occupying close spatial positions to each other [12] as words being indexed on Figure 4 below.



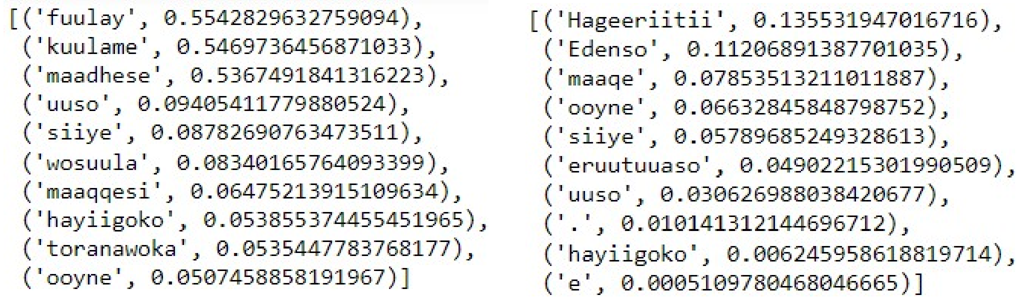
```
print("\n\n Tagged Corpus Tokenized Word index: \n", gensim_dictionary.token2id)
```

```
 Tagged Corpus Tokenized Word index:
 {'amaro': 0, 'kele': 1, 'koori': 2, 'nn': 3, 'worada': 4, 'adj': 5, 'aykesako': 6, 'beritaka': 7, 'bidzi': 8, 'biiro':
9, 'gaarda': 10, 'keexhuta': 11, 'modheenawo': 12, 'npl': 13, 'numc': 14, 'orijhe': 15, 'punc': 16, 'summita': 17, 'taamm
i': 18, 'vb': 19, 'vbp': 20, 'zaway': 21, 'zawitafa': 22, 'degexiyaase': 23, 'dmpron': 24, 'esafa': 25, 'fuulay': 26, 'h
a': 27, 'modhe': 28, 'nu': 29, 'orijhenawooko': 30, 'prpron': 31, 'wudey': 32, 'dandadha': 33, 'deexhona': 34, 'esaka': 3
5, 'hantuutusune': 36, 'hazaway': 37, 'kesese': 38, 'miijheko': 39, 'hantaxiiti': 40, 'kenge': 41, 'laga': 42, 'prep': 4
3, 'shahoko': 44, 'sumay': 45, 'vbf': 46, 'aalo': 47, 'adv': 48, 'afanuntey': 49, 'ayiiti': 50, 'busho': 51, 'eruwaso': 5
2, 'godo': 53, 'hate': 54, 'sconj': 55, 'weradaka': 56, 'ewoko': 57, 'maakesa': 58, 'na': 59, 'udi': 60, 'wegaka': 61, 'a
fanuntese': 62, 'hanta': 63, 'lcn': 64, 'sumaka': 65, 'xharey': 66, 'werguse': 67, 'wergusoko': 68, 'bushancce': 69, 'mii
nxe': 70, 'orijhena': 71, 'sumako': 72, 'worgusesa': 73, 'ato': 74, 'axii': 75, 'keexii': 76, 'keexutorosa': 77, 'maxxo':
78, 'mogeseko': 79, 'moroma': 80, 'oda': 81, 'sunge': 82, 'ababako': 83, 'addis': 84, 'axi': 85, 'ccn': 86, 'eyese': 87,
'goddoy': 88, 'hatee': 89, 'kaataara': 90, 'wona': 91, 'artidi': 92, 'axeko': 93, 'geede': 94, 'haydzhi': 95, 'yesa': 96,
'bidzunxoy': 97, 'erutesafa': 98, 'esuna': 99, 'gaacee': 100, 'hawudey': 101, 'hayxoy': 102, 'lakunxoy': 103, 'maake': 10
4, 'numo': 105, 'oyixoy': 106, 'qam': 107, 'suuseko': 108, 'utuma': 109, 'aniwoggan': 110, 'irpron': 111, 'kaxa': 112, 'l
am': 113, 'wogay': 114, 'yesenkeko': 115, 'zerre': 116, 'bargora': 117, 'silaara': 118, 'asiiwo': 119, 'aylera': 120, 'eh
idesakon': 121, 'esum': 122, 'gade': 123, 'keemonaraako': 124, 'makusune': 125, 'modhusi': 126, 'nagaadenaara': 127, 'nuu
ma': 128, 'nuunni': 129, 'osaaxi': 130, 'worguseesi': 131, 'wotera': 132, 'yeyiidi': 133, 'anguuzete': 134, 'basa': 135,
```

Figure 4. Word2Index Positioning Representation

3.4. Word2Vector and Feature Extraction

The plotting here below is sketched to show the distribution of words with their semantic similarity vectors using real numbers in the vector space. This means the graph tells us words internally are represented in numbers instead of words in the vector space. Word2Vec is demonstrated on the Skip-gram algorithm as shown on the figure 4 by extracting word features in real numbers, and by sketching the words-to-words distance of word vector generation. This practice is the key part of this study to handle the syntactic and semantic similarity of words based on the cosine similarity distance to distribute words in vector space. The word features below on figure 7 is about [*fuulay, kuulame, maadhese*] syntactic and semantic similarity to each other with a sample taken for testing purpose from the whole given KPT corpus. Hence, it is shown below here the most similarity to each other from, or among the given sample word for demonstration. Standing from these word features real number generation, it is simple to distribute words on the vector space mapping having these distance differences. The limited words are taken from the whole KPT corpus for the sake of training and testing. Therefore, the figure 5 below discusses about word features realization to vectored real numbers.



```
[('fuulay', 0.5542829632759094),
 ('kuulame', 0.5469736456871033),
 ('maadhese', 0.5367491841316223),
 ('uuso', 0.09405411779880524),
 ('siiye', 0.08782690763473511),
 ('wosuula', 0.08340165764093399),
 ('maaqqesi', 0.06475213915109634),
 ('hayiigoko', 0.053855374455451965),
 ('toranawoka', 0.0535447783768177),
 ('ooyne', 0.0507458858191967)]
```

```
[('Hageeriitii', 0.135531947016716),
 ('Edenso', 0.11206891387701035),
 ('maaqe', 0.07853513211011887),
 ('ooyne', 0.06632845848798752),
 ('siiye', 0.05789685249328613),
 ('eruutuuaso', 0.04902215301990509),
 ('uuso', 0.030626988038420677),
 ('.', 0.010141312144696712),
 ('hayiigoko', 0.006245958618819714),
 ('e', 0.0005109780468046665)]
```

Figure 5. Word to Vector representation for word features

The sketching below on Figure 6 depicts the reality of word embedding discussing the word to vector real number representation in the semantic similarity based on cosine distance measurement in the vector space distribution from the word features generated as above on Figure 5. This means the words are being distributed in the vector space based on the word features' distance measurement differences to evaluate the syntactic and semantic similarity in a given context of sentence. For example, the context similarity distance from Figure 5 above for [*fuulay, kuulame, maadhese*] have the nearest real number distance generation to each other but they are somewhat far apart in vector space embedding while measuring their syntactic and semantic similarity. These first two [*fuulay, kuulame*] are a noun part of speech and have more nearest distance real number generation in addition to this they have contextually similar meanings. So, they can replace each other in the same sentence. Figure 6 below is about word vector distribution in a vector space representation.

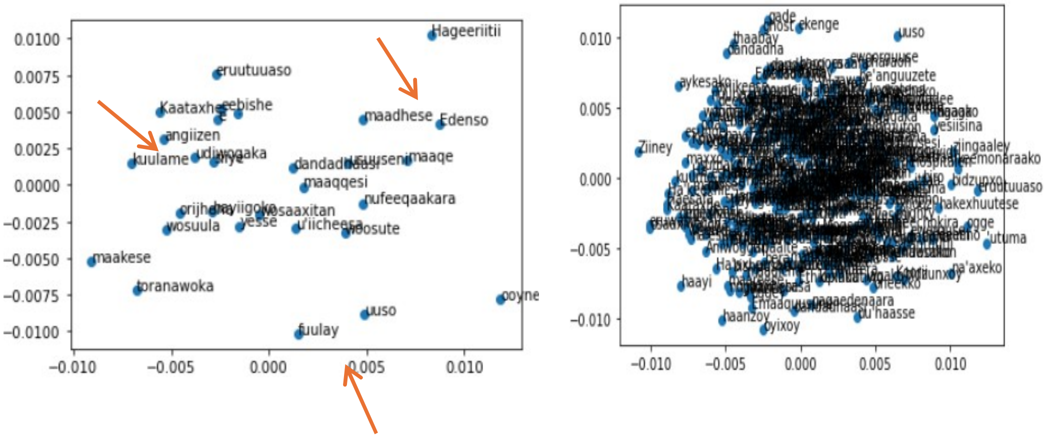
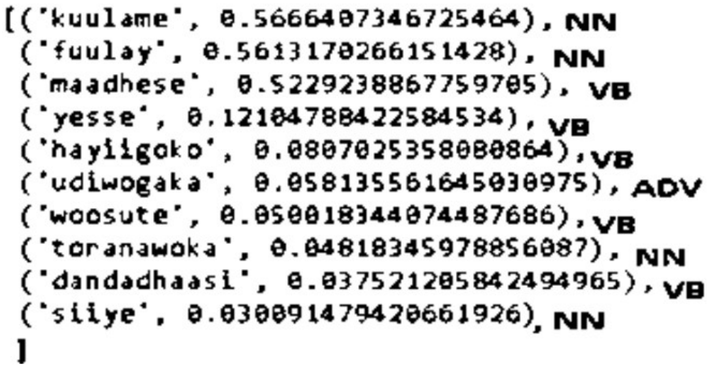


Figure 6. Semantic similarity vector space distribution

3.5. Word Features to POS Tagging

Features are the indivisible distinctive properties of input patterns that help in differentiating between the classifications of input patterns. After the extraction of word features, the corpus is separated into training data 90% and testing data 10% proportion ratio building a model for testimonial purposes. The feature extractor searches for the corresponding 'word features' to 'tags' in the corpus from the indices matching in 'word-tag-index' mapping as shown in Figure 7 on tokenized word indexes. These word indices are used for searching the words, its corresponding tags, and their word vector features with POS tags appended to them. The limited number of words limited the number of tagsets that were taken from the whole KPT corpus, that is important for the sake of training and testing. According to this taken sample size, there are only 10 words, which are being limited to only three POS tagging tagsets.



```
[('kuulame', 0.5666407346725464), NN
 ('fuulay', 0.5613170266151428), NN
 ('maadhese', 0.5229238867759705), VB
 ('yesse', 0.12104788422584534), VB
 ('hayiigoko', 0.0807025358080864), VB
 ('udiwogaka', 0.058135561645030975), ADV
 ('woosute', 0.050018344074487686), VB
 ('toranawoka', 0.04818345978856087), NN
 ('dandadhaasi', 0.037521205842494965), VB
 ('siiye', 0.030091479420661926), NN
 ]
```

Figure 7. Word feature extraction with its word POS

According to Figure 7 above, the research question [How Bi- LSTM RNN based neural word embedding approach evaluation could better perform concerning N-gram based POS tagging for Koorete language?], is done for the experimental demonstration by extracting word vector features to perform neural word embedding POS tagging. Based on this question, the Koorete language is being demonstrated for this research contribution in both preparing the KPT corpus from scratch and developing the POS tagger on empirical evaluation of Bi-LSTM and N-gram language models. The same titled research was done for Amharic POS tagging using neural word embedding in Ethiopia [8]; other many researches are also studied on abroad languages. The abroad languages research has demonstrated using this deep neural word embedding model as the state-of-the-art algorithm for POS tagging to achieve better performance on the sequence labeling tasks [9, 2, 3, 6]. Also, the N-gram-based statistical approach is demonstrated for the relative performance evaluation comparison to the neural one.

3.6. Results and Discussion

The experiment demonstration fed the same data size for SimpleRNN, LSTM, and Bi_LSTM at the time of training to distinguish the performance difference in accuracy score, and also fed the same data size at the time of testing the models. The experiment results recorded are the following accuracy scores as shown in Table 4. In aspects of Bi-LSTM, there is a slight difference in accuracy occurrence for training and testing data in the interval of 10 epoch modeling. However, there is a big difference on empirical summary value between training and testing accuracy for Trigram tagger and Bi-LSTM described in Table 4 below.

Table 4. Summary of Accuracy for both training and testing data

№	Model		Accuracy of training	Accuracy of testing
1	SimpleRNN		73.99 %	72.60%
2	LSTM RNN		98.50%	98.45%
3	Bi-LSTM RNN		98.53%	98.49%
4	Unigram		96.34%	76.81%
5	Bigram		97.10%	76.84%
6	Trigram tagger		97.21%	77.29%

In this study, the number of N-grams used are default, regular expression, unigram, bigram, and trigram taggers using backoff parameters. It is easy to conclude the performance of each N-gram from the above table 3 to compare their accuracy achievement of both training and testing. As it is scrutinized the unigram and the bigram performed quite less achievement than the trigram realization. Besides, the Trigram tagger showed us the value of a big difference between the accuracy of training data and testing data on N-gram based tagger. But when comparing the N-gram performance versus the Bi_LSTM performance determination, the Trigram tagger model achieved a less effective score than the Bi_LSTM RNN model for both training data and testing data. Bi_LSTM model performed about 98.53% and 98.49% scores which is somewhat nearer approximation for accuracy of both training and testing data, respectively. Also, the Bi_LSTM model achieved a more effective score value than the N-gram-based statistical tagging approach. This performance comparison demonstrates that the truth of the Bi_LSTM RNN word embedding POS tagging approach performs better than the N-gram statistical POS tagging approach. This truth is supported by Peilu Wang et al. [2] as Bi_LSTM RNN neural word embedding is a state-of-the-art performance of 97.40% tagging accuracy on Penn Treebank POS tagging. Besides, Erick Fonseca et al. [5] presented Portuguese POS tagging word embedding with the achievement of state-of-the-art performance with 97.57% overall accuracy.

4. Conclusion and Future Works

4.1. Conclusion

This research study testing is implemented on the Koorete language. The language is spoken by the Koore people who are located at Koore Zone former Amaro special woreda, South Regional State of Ethiopia, and it has about 37-Latin script. The language category falls under Ometo language family [18]. It has about 350,000 speakers (Awoke, Koore People, 2020). The corpus is collected from Koorete dictionary, Matthew Gospel of the New Testament Bible in Koorete, Book of '**Dicchoo**' published in 2006 E.C, and the Koorete language department at Dilla College of Teacher Education.

This study used the empirical evaluation of Bi-directional LSTM RNN based neural word embedding concerning N-gram based statistical POS tagging approaches to answer the research question how better to develop Koorete POS tagging. For this reason, KPT corpus is used size of 33220 words with 24 tagsets, and then divided this corpus into 90% training data and 10% testing data. The experiment on Bi-LSTM RNN word embedding POS tagging approach could be performed better than the N-gram based statistical POS tagging approach with an accuracy of 98.53%. This shows Bi_LSTM neural word embedding has better potential on training and testing data accuracy. This truth is supported by Peilu Wang et al. [2] as Bi-LSTM RNN neural word embedding is a state-of-the-art performance of 97.40% tagging accuracy is achieved on Penn Treebank POS tagging. Besides, Erick Fonseca et al. [5] presented Portuguese POS tagging word embedding with achievement of the state-of-the-art performance with 97.57% overall accuracy.

4.2. Future Works Enhancement

From the standing point of the above conclusion, the Koorete language POS tagger is evaluated using deep learning neural word embedding, and the N-gram based statistical approaches for developing other high-level NLP applications. Other researchers can develop Koorete high-level NLP applications beyond POS tagging such as syntactic parsing, word-sense disambiguation, named-entity recognition, machine translation, morphological analysis, information extraction, …, etc. based on this POS tagging application as a pre-condition. Because POS tagging is a pre-requisite for these aforementioned NLP applications, this study has implemented the foot-step for advanced NLP applications. Other researchers can use SimpleRNN, ANN, and LSTM neural word embedding based POS tagging approach because these approaches have a competing performance with Bi-LSTM neural word embedding as features.

Declarations

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Availability of data and materials

The research data are available from the authors and can be accessed from the authors accordingly.

Competing interests

The author declares that no competing interest.

References

[1] Sintayehu Hirpassa, G.S. Lehal. "Improving part-of-speech tagging in Amharic language using deep neural network". Adama Science and Technology University, Ethiopia, and Punjabi University, India. Elsevier Ltd: www.cell.com/heliyon. June 9, 2023. https://doi.org/10.1016/j.heliyon.2023.e17175

[2] S.Ezhilarasi, Dr.P.Uma Maheswari. “Designing the Neural Model for POS Tag Classification and Prediction of Words from Ancient Stone Inscription Script”. Anna University, India. International Journal of Aquatic Science. ISSN: 2008-8019. Vol 12, Issue 03, 2021

[3] Wubetu Barud Demilie. “Analysis of implemented part of speech tagger approaches: The case of Ethiopian languages”. https://doi.org/ 10.17485/IJST/v13i48.1876 30/12/2020. IST, India.

[4] Diksha Khurana1, Aditya Koli1, Kiran Khatter1,2 and Sukhdev Singh1,2. “ Natural Language Processing: State of The Art, Current Trends and Challenges” Manav Rachna International University 19 Jun 2018

[5] Senait Gebremichael Tesfagergish. “Part-of-Speech Tagging via Deep Neural Networks for Northern-Ethiopic Languages”. http://dx.doi.org/10.5755/j01.itc.49.4.26808. 2020

[6] Bin Wang_, Student Member, IEEE, Angela Wang_, Fenxiao Chen, Student Member, IEEE, Yuncheng Wang and C.-C. Jay Kuo, Fellow, IEEE. “Evaluating Word Embedding Models: Methods and Experimental Results”. arXiv:1901.09785v2 [cs.CL]. Jan. 29, 2019

[7] Pinky Sitikhu1, Kritish Pahi2, Pujan Thapa3, Subarna Shakya4.“A Comparison of Semantic Similarity Methods for Maximum Human Interpretability.” arXiv:1910.09129v2 [cs.IR]. 31 Oct 2019. Tribhuwan University.

[8] Mequanent Argaw. “Amharic Parts-of-Speech Tagger using Neural Word Embeddings as Features.” Addis Ababa Institute of Technology School of Electrical and Computer Engineering, AA (January, 2019).

[9] Anastasyev D. G., Gusev I. O., Indenbom E. M. “Improving Part-of-Speech Tagging via Multi-task Learning and Character-level Word Representations.” ABBYY, Moscow Institute of Physics and Technology, Moscow, Russia Moscow, May 30—June 2, 2018

[10] NgoXuanBacha,b,*,NguyenDieuLinha,TuMinhPhuonga,b . “An empirical study on POS tagging for Vietnamese social media text”. Science Direct Computer Speech & Language50 (2018)1_15 www.elsevier.com/locate/csl

[11] Mariya Koleva, Melissa Farasyn, Bart Desmet, Anne Breitbarth and Véronique Hoste. “An automatic part-of-speech tagger for Middle Low German.” (2018) Ghent University

[12] MD. Asif Iqbal, Omar Sharif, Mohammed Moshiul Hoque, Iqbal H. Sarker. “Word Embedding based Textual Semantic Similarity Measure in Bengali”. Volume 193, 2021, Pages 92-101. https://doi.org/10.1016/j.procs.2021.10.010

[13] Amitha Mathew[1], P.Amudha[2] and S.Sivakumari[3]. “Deep Learning Techniques: An Overview.” https://www.researchgate.net/publication/341652370. (August 2, 2021) Avinashilingam University

[14] Rkia Bani , Samir Amri , Lahbib Zenkouar , Zouhair Guennoun. “Deep Neural Networks for Part-of-Speech Tagging in Under-Resourced Amazigh”. Vol. 37, No. 3, June 2023, pp. 611-617. https://doi.org/10.18280/ria.370310 Mohammed V University,Rabat 10090, Morocco

[15] Chengwei Wei1 , Runqi Pang1 , and C.-C. Jay Kuo1. “GWPT: A Green Word-Embedding-based POS Tagger”. arXiv:2401.07475v1 [cs.CL] 15 Jan 2024. University of Southern California, Los Angeles, California, USA.

[16] Elham Mohammadi, Hessam Amini and Leila Kosseim. “Neural Feature Extraction for Contextual Emotion Detection” Proceedings of Recent Advances in Natural Language Processing, pages 785–794, Varna, Bulgaria, Sep 2–4, 2019. https://doi.org/10.26615/978-954-452-056-4_091

[17] Samuel Zinabu Haile, Binyam Sisay Mendisu. “Examining Teachers' Practice of Phonological Awareness (PA) in Early Grades: A Qualitative Study of Koorete Language Classes, Southern Ethiopia”. Addis Ababa University and The Africa Institute, Global Studies University, Sharjah. Journal of Ethiopian Studies, 2024 - ejol.aau.edu.et

[18] BELETU REDDA. “THE MORPHOLOGY OF KOORETE: Koorete Verb Morphology”. Addis Ababa, Addis Ababa University. http://localhost:80/xmlui/handle/123456789/6353

[19] Ambrose Bangnia. “Challenges of the Teaching and Learning of French as a Foreign Language in Ghana: The Way Forward.” University of Education, Winneba-Ghana Volume 4 Issue 01, January 2020

[20] Khwlah Alrajhi1 and Mohammed A ELAffendi2.“ Automatic Arabic Part-of-Speech Tagging: Deep Learning Neural LSTM Versus Word2Vec”. ISSN (2210-142X) Int. J. Com. Dig. Sys. 8, No.3 (May-2019) Prince Sultan University, Riyadh, Saudi Arabia

[21] Serkan Ballı1, Onur Karasoy1”. Development of content-based SMS classification application by using Word2Vec based feature extraction”. ISSN 1751-8806 Accepted on 15th October 2018, Turkey doi: 10.1049/iet-sen.2018.5046 www.ietdl.org

[22] Birhanesh Fikre Shirko. “Part of Speech Tagging for Wolaita Language using Transformation Based Learning (TBL) Approach”. Volume 10 Issue No.9 Wolaita Sodo University, Ethiopia. IJESC, September 2020.

[23] Pinky Sitikhu1, Kritish Pahi2, Pujan Thapa3, Subarna Shakya4. “A Comparison of Semantic Similarity Methods for Maximum Human Interpretability”. arXiv:1910.09129v2 [cs.IR] 31 Oct 2019

[24] NgoXuanBacha,b,*,NguyenDieuLinha,TuMinhPhuonga,b. “An empirical study on POS tagging for Vietnamese social media text”. www.elsevier.com/locate/cs Computer Speech & Language 50(2018) 1_15, Vietnam

[25] Jabar H. Yousif. “Hidden Markov Model Tagger for Applications Based Arabic Text: A review” International Journal of Computation and Applied Sciences IJOCAAS, Volume 7, Issue 1, August 2019, ISSN: 2399-4509

[26] Tej Bahadur Shahi, Tank Nath Dhamala, Bikash Balami. “Support Vector Machines based Part of Speech Tagging for Nepali Text”. Volume 70– No.24, May 2019

[27] Sam Goundar. “Research Methodology and Research Method” Victoria University of Wellington March 2012. https://www.researchgate.net/publication/333015026

[28] Binyam Sisay Mendisu. “ASPECTS OF KOORETE VERB MORPHOLOGY” Master of arts in linguistics, Oslo University. March, 2008

[29] PANTULKAR SRAVANTHI1, DR. B. SRINIVASU2. “SEMANTIC SIMILARITY BETWEEN SENTENCES” Volume: 04 Issue: 01 | Jan -2017. www.irjet.net. p-ISSN: 2395-0072. Stanley College of Engineering and Technology for Women, Telangana- Hyderabad, India

[30] Wahyudin Darmalaksana, Wildan Budiawan Zulfikar. “Latent semantic analysis and cosine similarity for hadith search engine”. UIN Sunan Gunung Djati Bandung. February 2020.

[31] Wilson Gonzalo Rojas Yumisaca, Nancy Georgina Rodríguez Arellano, Nanci Margarita Inca Chunata, María Guadalupe Escobar Murillo. “Articulatory Phonetics In The English Languaje Pronunciation Development” Magister En Docencia Universitariay 2018.

[32] Duressa Tamirat Gemeda. “Corpus based Auto-Completion of N-GRAM WORD PREDICTION FOR AFAN OROMO WORDS”. Adama, Ethiopia Septemper, 2016

[33] Saranlita Chotirat, Phayung Meesad. “Part-of-Speech tagging enhancement to natural language processing for Thai wh-question classification with deep learning”. Volume 7, Issue 10, October 202. https://doi.org/10.1016/j.heliyon.2021.e08216.

[34] Kåre Hartlapp Lærum. “A study of Machine Learning for Predictive Maintenance” Norwegian University of Science and Technology. June 2018

[35] Md. Mostafizer Rahman, Yutaka Watanobe and Keita Nakamura. “A Bidirectional LSTM Language Model for Code Evaluation and Repair”. Symmetry 2021, 13, 247. University of Aizu, Aizu-Wakamatsu. https://doi.org/10.3390/sym13020247.

[36] NUNIYAT KIFLE ABEBE’. “WORD SEQUENCE PREDICTION FOR AMHARIC”. February 2011 Addis Ababa University, Ethiopia

[37] Grigori Sidorov1, Francisco Velasquez1, Efstathios Stamatatos2, Alexander Gelbukh1, and Liliana Chanona-Hernández3. “Syntactic N-grams as Machine Learning Features for Natural Language Processing1”. Mexico, University of the Aegean,Greece MICAI 2012. www.cic.ipn.mx/~sidorov

[38] S. Girma and W. Zenebe, "English - Koorete – Amharic School Dictionary (Ingiliizete - Koorete -Qawete Erunxi Zawa Kato Tuula)," p. 10(vi), 2022. https://www.sil.org/system/files/reapdata/10/48/21/104821441593609927722414943761089651147/English_Koorete_Amharic_School_Dictionary_2022.pdf, and, Koore Zone, former Amaro Special Wereda Culture, Tourism, and Communication Office Annual Report: “Dicchoo” Book, 2006 E.C.