Nepali POS Tagging Using Deep Learning Approaches

Sarbin Sayami and Subarna Shakya*

Central Department of Computer Science and IT, Institute of Engineering, Tribhuvan University, Kathmandu, 44600, Nepal

*Corresponding author. E-mail: drss@ioe.edu.np

ABSTRACT

Deep Learning approaches are being extensively used in Part of Speech (POS) tagging. POS tagging is one of the important step in Natural Language Processing (NLP) including Machine Translation, Retrieval of Information, developing question answering system, word sense disambiguation, text summarization, Named Entity Recognition, text to speech conversion and classification. The efficiency of POS tagging heavily rely on syntactic, contextual information and morphology of the language. POS tagging in Nepali Language is very difficult as it is morphologically rich. This research paper focuses on implementing and comparing various deep learning approaches for POS tagging in Nepali Language. Recurrent Neural Network (RNN), Gated Recurrent Unit (GRU), Long Short Term Memory (LSTM) and Bidirectional LSTM were implemented in tagged Nepali corpus. The result of Bidirectional LSTM (Bi-LSTM) was better than other approaches.

Keywords: POS, NLP, RNN, GRU, LSTM, Bi-LSTM, Nepali corpus

INTRODUCTION

NLP belongs to domain of Artificial Intelligence which deals with techniques to make computer smarter enough to understand words and sentences in a way we human understands it. The concept was introduced in 1950s. A research paper "Machine and Intelligence" had the same concept and was published by Alan Turing. Similarly, around 1960's SHRDLU was introduced that worked in restricted "block worlds".

POS tagging can be carried out using different techniques like symbolic, statistical and artificial neural network. The first one i.e. symbolic technique is highly dominated by linguistic rules which again requires thorough understanding of the schemes required for representation of knowledge in an unambiguous way. Statistical technique can only work on the availability of sufficiently large text corpora.

These corpora are created by collecting real examples which reflects the linguistic phenomena. These corpora are then utilized to form approximate models which would be able to represent linguistic phenomena without the involvement of hard linguistic rules. Statistical techniques are prominent in recognition of speech,

Article history:
Received 10 March 2020; Received in revised from 6 October 2020;
Accepted 3 November 2020; Available online 11 November 2020

POS tagging, machine translation, acquisition of lexical units of language form, collocations, learning of grammars etc. The last one i.e. artificial neural network emphasizes on creation of massively parallel network where statistical techniques are blended with other standard techniques for representing knowledge. These networks are trained where formally weights are adjusted that will aid in transformation, inference and development of modified logic rules and formulae.

POS tagging is a process of searching grammatical of lexical patterns. It also deals with disambiguation of word-category. This is carried out by tagging every word in sentences. The most commonly used POS tags are nouns, adverbs, verbs, adjectives (Ahmad, 2017).

POS tagging can be considered as one of the basic step in NLP. It can affect the quality in developing applications related to question answering system, recognizing speech, machine translation from one language to another language, information retrieval and word sense disambiguation (Rana, 2010).

POS tagging are based on numerous well established techniques like rule based, statistics, and neural network. The underlying focus of these techniques are to provide if possible a correct single POS tag within a specified context. If not then it will try to assign the most likely POS among the available tags (Ortiz, 2001).

A. Rule Based

Linguistic rules are developed from the hard core knowledge of linguistic. These rules are used to label a word within the given context unambiguously (Antony, 2010). Since these types of linguistic rules are based on context, they are also considered as context frame rules.

The tags are assigned to words in the sentences on the basis of highest occurrence of tag for the selected word.

In 1992 Eric Brill developed a tagging program known as Brill Tagger which was highly influenced by the concept of regulation and transformation, therefore also known as transformation-based tagger. The fundamental concept of Brill tagger is to calculate the errors of the tagged words on the basis of scores. These errors are then corrected by implementing chain of rules.

Brill tagger has three components. The first component focuses on initialization of known and unknown words. The second component takes care of calculation of errors after the implementation of chain of rules. These errors are then minimized by the concept of learning so that the best rules are generated. The third component deals with the selection of generated rules which satisfies the threshold posed. Lexicon, Lexical and contextual annotation are carried out by Brill Tagger (Fahim, 2006).

B. Stochastic Approach

Frequency, probability or statistics forms the basis of stochastic technique. Frequency based technique rely on the highest frequency count of tag for a given word in the annotated text within the specified context to tag the word in an unannotated text.

One of the example of statistical tagging is Unigram which works on the basis of single token only. It works with reference to the single word as its context and

determines POS tags (Fahim, 2006). Unigram focuses on developing context model from the available repository of tagged sentences.

The tag set generated from Unigram tagger is used for prior training purpose where labelling is carried out on the basis of most frequently occurring tags. These tags are checked against the tags stored in the dictionary of the tagger. 'None' labelled tag set will not be found in training set (Fahim, 2006).

Best tag is found out after calculation of probability of occurrence with respect to previous tags, such concept is shared by Hidden Markov Model (HMM). It is a statistical model developed by L.E. Baum and coworkers. It is based on Markov process where the concept of hidden states are used. It is highly used in speech recognition and bioinformatics.

It chooses the tag sequence by maximizing the following formula:

P(word|tag) * P (tag|previous n tag)

HMM follows a different approach in comparison to the existing POS taggers. It focuses on combining best tags for word order in contrast to other taggers where greedy method is implemented to tag a word at a time. HMM also takes care of optimal combination. The probability of most likely sequences of POS tags can be found out by using the following formula

$$Pr\ (t_{1,n}, w_{1,n}) \approx \Pi_{i=1}^{n} \quad (Pr\ (\ t_i \mid t_{i-k,\ i-1}) \ X\ Pr(w_1 \mid t_i)$$

where

{w1, w2… ww} is a set of words,

{t1, t2… tT} is a set of POS Tags,

W_1, n = W_1 W_2 … W_n is a sentence of n word.

The probability of current tag t1 is dependent on k previous tags and the probability of current word depends on the existing current tag t1 (Fahim, 2006).

Though unigram and HMM based taggers are easy to develop, it is difficult to integrate complex features into them. Therefore, to eradicate this problem of probability models, Maximum Entropy (ME) based tagger are introduced which provides methodologies and techniques for integration of complex features.

Provided sentences, ME based tagger calculates the conditional probabilities of tag sequences $t_1 \ldots t_n$ as:

$$P_r\ (t_{1\ \ldots\ t_n} \mid w_{1\ \ldots\ w_n}) \approx \Pi_{i=1}^{n} \quad P\ (\ t_i \mid C_i)$$

where C_1, … C_n are the corresponding contexts for each word appearing in the sentence. The context 'C' of a word 'W' incorporates all the previous assigned tags before 'W'.

ME based taggers are based on concept of features. Features can be considered as binary valued functions which are used to represent constraints. These features are used to encode elements of context 'C' so that prediction of tag t of word W can be carried out.

ME tagger uses the concept of features to calculate $P(t_i|C_i)$. It adjust the weights of the features by the help of training corpus so that entropy of the probabilistic model can be maximized (Fahim, 2006).

C. Neural Network Approach

Apart from feed forward network, several variations of RNN have been implemented for POS tagging. Some of them are Simple RNN, GRU, LSTM and bidirectional RNN's (Firoj, 2016; Archit, 2018).

D. Problem Statement

Nepali Language can be considered as a low resource language. The experiments carried out therefore do not comply with the state of the art performance with other high resource languages.

Similarly, since Nepali language is morphologically rich, rule based and statistical techniques do not show significant results as they do not take care of context and sequence.

Likewise very few implementations of deep learning approaches can be found with respect to Nepali Language.

E. Objective

To implement deep learning approaches like Simple RNN, GRU, LSTM and Bi-LSTM for POS tagging of Nepali text and to compare the results of these approaches. The main objective is to increase the performance of tagging.

LITERATURE REVIEW

RNN is vibrant in modeling sequential data. It has been successfully tested for tagging sequential inputs. Therefore, its use in NLP is increasing day by day. LSTM is a RNN proposed by Hochreiter and Schmidhuber. These LSTM units are implemented to propagate salient features of input sequence over a long distance. This solves the problem of long distance dependencies as it is capable of retaining salient features.

The performance of LSTM networks are better while dealing with long word dependencies in comparison to other learning algorithms. The POS tagger based on LSTM have shown state-of-the-art performances when tagging text at character level and at word level (Wang, 2015).

In (Peilo, 2015), Bi-LSTM was used to represent word in vectors after reading the character sequences that formed each word. Bi-LSTM is associated with vectors and LSTM parameters. Each vector is associated with each character type and LSTM parameters is associated with encoding of idiosyncratic lexical and morphological knowledge. The evaluation of the model was carried out using vector based and language based models. They experimented these models on several languages as well. Their results show that their model obtained the state-of-the art performance on POS tagging and establishing the new best performance in the English language.

In (Yushi, 2016), bi-directional RNN with LSTM units were used for segmenting Chinese words. The segmentation was used as preprocessing to model

Chinese sentences and articles. Bi-LSTM does not require any prior information, hand craft features from context and predesigns in comparison to classical methods.

Bi-LSTM is also capable of retaining and propagating contextual information in both directions. Their results depict state-of-the-art performance in segmenting words on both traditional and simplified Chinese dataset. They implemented Bi-LSTM for capturing long distance dependencies via bidirectional links.

In (Greeshma, 2018), deep learning algorithms like RNN, GRU, LSTM, Bi-RNN, Bi-GRU and Bi-LSTM were implemented for sequence to sequence modeling. Their results show that Bi-RNN, Bi-GRU and Bi-LSTM performed well which was verified by calculation of loss function using binary cross entropy.

They also found out that the accuracy of the system was directly proportional to the size of the word embedding vector. The accuracy obtained by RNN was 91.68%, LSTM was 91.74%, GRU was 91.66 and Bi-LSTM was 92.66 respectively.

BACKGROUND STUDY

A. Simple RNN

In simple RNN architecture, multiplication of Input (Xt) and previous output (ht-1) is carried out. It can be seen from Figure 1. This output is passed through Tanh activation function to get the final output. There are no Gates present in the architecture.

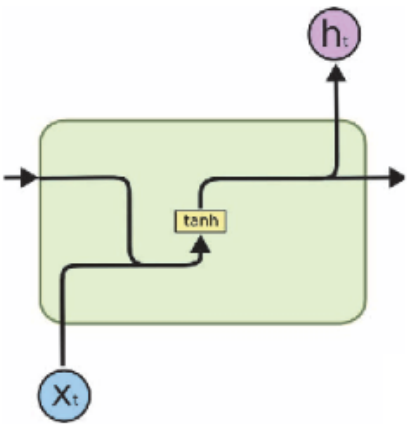

Figure 1 Simple RNN Architecture

B. GRU

Update gate and forget gate are introduced in the GRU architecture. Update gate decides whether or not to propagate the previous output h_{t-1} to next cell as ht. Similarly, forget gate is responsible for dealing with weights W_t as shown in Figure 2.

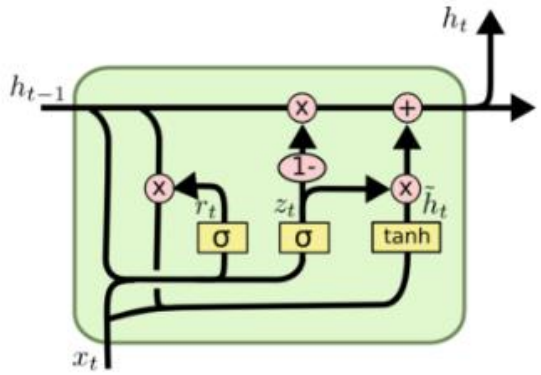

Figure 2 GRU Architecture

The main equations used are as follows

$z_t = \sigma\ (W_z\ .[h_{t-1}, x_t])$

$r_t = \sigma\ (W_r\ .[h_{t-1}, x_t])$

$\check{h}_t = \tanh\ (W.[\ r_t * h_{t-1}, x_t])$

$h_t = (1 - z_t) * h_{t-1} + z_t * \check{h}_t$

C. LSTM

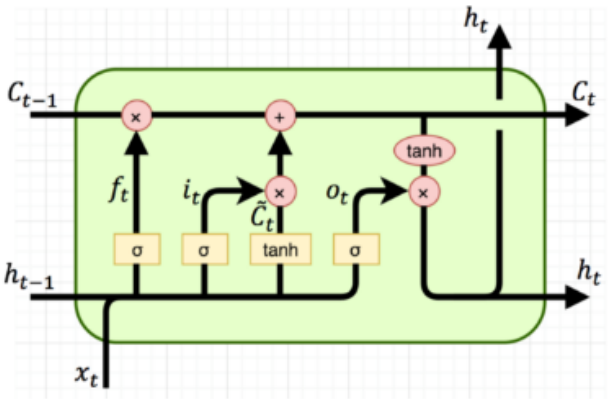

Figure 3 LSTM Architecture

Therefore, LSTM architecture consists of two math operations for dealing with new sets of weights. Its architecture is shown in Figure 3.

The main equations used are as follows

$$f_t = \sigma\,(W_f\,.[h_{t-1}\,,\,x_t] + b_f)$$

$$i_t = \sigma\,(W_i\,.[h_{t-1}\,,\,x_t] + b_i)$$

$$\check{c}_t = \tanh\,(W_c\,.[h_{t-1}\,,\,x_t] + b_c)$$

$$Ct = f_t * C_{t-1} + i_t * \check{c}_t$$

$$o_t = \sigma\,(W_o\,.[h_{t-1}\,,\,x_t] + b_o)$$

$$h_t = o_t * \tanh\,(C_t)$$

D. Bidirectional Neural Network

Bi-LSTM neural network are constructed from LSTM units which can retain long distance dependencies and can propagate information in both directions. It does not consider redundant context information while learning long term dependencies. Since Bi-LSTM performance is extremely better on sequential modeling, it has been implemented in various NLP tasks.

Bi-LSTM neural network is able to retain contextual features from past and future. Similarly, it is also capable of capturing information of sequential dataset. Bi-LSTM consists of two parallel layers which propagates information in two directions. These two layers form the basis for memorizing information of sentences from both directions.

RESEARCH METHODOLOGY

A. Data Collection

Data were collected from Madan Puraskar Pustakalaya. It consists of Nepali English parallel corpus annotated with 40 POS tags developed and contains nearly 88000 words.

The design of this Nepali POS Tag-set was inspired by the PENN Treebank POS Tag-set. Hence, whenever possible, the same naming convention has been used as in the case of the Penn Treebank Tag-set. The sample of POS tagged Corpus is as shown in Figure 4.

हामी<**PP**>			कसै<**DUM**>ले<**PLE**>
अस्बेस्टस<**NNP**>मा<**POP**>		आपत्तिजनक<**JJ**>	गुण<**NN**>
रहेको<**VBKO**>			सुन्नु<**VBI**>भन्दा<**VBO**>
अगाडि<**RBO**>	वर्षौं<**NN**>		अघि<**POP**>को<**PKO**>

बारेमा<**POP**> कुरा<**NN**> गरिरहेका<**VBKO**> छौँ<**VBX**> ।<**YF**>

Figure 4 Sample of Nepali POS tagged corpus

The definition of tags shown in Figure 4 are as follows

Noun - N.*
Adjective - J.*
Pronoun - P.*
Determiner - D.*
Verb - V.*
Adverb - R.*
Postposition - I.*
Number - M.*
Particle – TT

B. Data Preparation

Parallel corpus was developed where first consisted of plane Nepali text and other consisted of only tags as shown in Figure 5.

हामी कसै ले अस्बेस्टस मा आपत्तिजनक गुण रहेको सुन्नु भन्दा अगाडि वर्षौँ अघि को बारेमा कुरा गरिरहेका छौं।
<PP> DUM> <PLE> <NNP> <POP> <JJ> <NN> <VBKO> <VBI> <VBO> <RBO> <NN> <POP> <PKO> <POP> <NN> <VBKO> <VBX> <YF>

Figure 5 Sample of plane and its corresponding tags

C. Training

The training was carried out using architecture as shown in Figure 6. This architecture was used for training as it does not require fixed input data and is able to reach future input information from the current state. Simple RNN, LSTM, GRU and Bi-LSTM were implemented. Bi-LSTM was chosen due to its ability to understand context better from both directions.

The two hidden layers of opposite directions are connected to the same output.

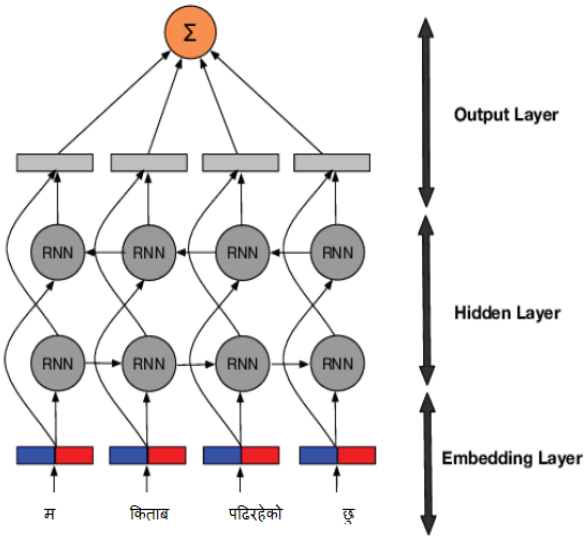

Figure 6 Architecture for Training

D. Testing

The model was tested using the testing set. The data set was compartmentalized into three sections i.e. training, development and testing. The instances for training, development and test were 1341, 191 and 383 respectively.

i. Architecture

Table 1 Architecture of RNN

Layer (type)	**Output Shape**
Embedding	100, 25
Simple RNN/ LSTM/ Bi-LSTM/ GRU	100, 50
TimeDist	None, 100, 40
TimeDist	None, 100, 40

As shown in Table 1, the architecture consists of 4 layers. The first one is the embedding layer, followed by variants of RNN's and finally there are two distributed layers. The final output layer consists of 40 units as there are 40 tags defined and output has to be one of them.

ii. Vocabulary Size

The generated vocabulary size was 6850 with a maximum sequence length of 100. The untokenized word was set to '_unk_' as shown in Figure 7.

```
VOCAB_SIZE = 6850
MAX_SEQUENCE_LENGTH = 100
UNK_TOKEN = '__unk__'
```

Figure 7 Setting of vocabulary size and maximum sequence

iii. POS tags

A total of 40 POS tags were used and were given integer numbers as shown in Table 2.

iv. Preparation of Text and Labels in List

The data was loaded and was preprocessed as shown in Figure 8.

```
[(['\ufeff६१', ' वर्षीय', ' पियरे', ' भिन्केन', ' नोभेम्बर', ' २९', ' बाट', ' सल्लाहकार', 'को', ' रूप', ' मा', ' सञ्चालक', ' समिति', 'मा', ' आउनुहुनेछ', ' ।'], [1, 2, 3, 3, 3, 1, 4, 5, 6, 5, 4, 5, 5, 4, 7, 8])]
```

Figure 8 Text Preparation

As shown in Figure 8, the first half represents the tokens and the second half represents their POS with their respective integer values. The complete integer values for POS tags are shown in Table 2.

v. Preparation of train and test data

The data were divided for training, development and testing. The instances generated after executing the program for training, development and test were 1341, 191 and 383 respectively as shown in Figure 9.

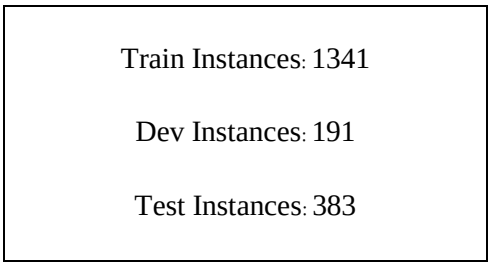

Figure 9 Division of data

vi. Preparation of vocabulary

The vocabulary was prepared based on the occurrences of tokens. It is based on most frequent tokens, least frequent tokens and unknown tokens. It can be seen from Figure 10.

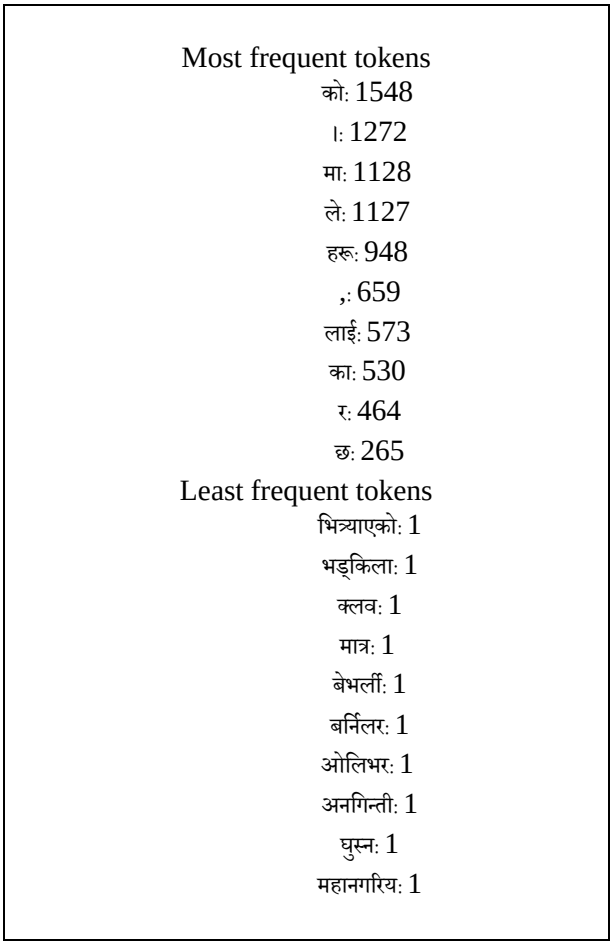

Figure 10 Sample of most frequent and least frequent tokens

The unknown tokens are then filtered out to reduce the memory consumption. The total numbers of unknown tokens are listed out in Figure 11.

Train: 0/34082

Dev: 558/4642

Test: 1321/10226

Figure 11 Total number of unknown tokens filtered

vii. Simple RNN POS tagger

The summary of the model is as shown in Table 3.

Table 3 Simple RNN Summary

Layer (type)	**Output Shape**	**Param #**
Embedding	(None, 100, 25)	171300
Simple RNN	(None, 100, 50)	1275
TimeDist	(None, 100, 40)	1040
TimeDist	(None, 100, 40)	0

Total params: 173,615

Trainable params: 173,615

Non-trainable params: 0

The accuracy obtained after training is 85.04% with a loss function of 0.5662. The test accuracy obtained for simple RNN was 96.84%.

viii. Unidirectional LSTM based POS tagger

The summary of the model is as shown in Table 4.

Table 4 Unidirectional LSTM Summary

Layer (type)	**Output Shape**	**Param #**
Embedding	(None, 100, 25)	171300
LSTM	(None, 100, 50)	5100
TimeDist	(None, 100, 40)	1040
TimeDist	(None, 100, 40)	0

==========

Total params: 177,440
Trainable params: 177,440
Non-trainable params: 0

The accuracy obtained after training is 98.97% with a loss function of 0.0612. The test accuracy obtained for unidirectional LSTM was 96.48%.

ix. Bidirectional LSTM based POS tagger Neural Network

The summary of the model is as shown in Table 5.

Table 5 Bidirectional LSTM Summary

Layer (type)	Output Shape	Param #
Embedding	(None, 100, 25)	250050
Bidirection	(None, 100, 50)	10200
TimeDist	(None, 100, 40)	2040
TimeDist	(None, 100, 40)	0

==========

Total params: 183,540
Trainable params: 183,540
Non-trainable params: 0

The accuracy obtained after training is 99.62% with a loss function of 0.0190. The test accuracy obtained for bidirectional LSTM was 97.27%.

x. GRU based POS tagger

The summary of the model is as shown in Table 6.

Table 6 GRU Summary

Layer (type)	**Output Shape**	**Param #**
Embedding	(None, 100, 25)	171300
GRU	(None, 100, 50)	3825
TimeDist	(None, 100, 40)	1040
TimeDist	(None, 100, 40)	0

==========

Total params: 176,165
Trainable params: 176,165
Non-trainable params: 0

The accuracy obtained after training is 99.60% with a loss function of 0.0195. The test accuracy obtained for GRU was 96.86%.

xi. Results of Nepali POS tagging

Simple RNN, LSTM, and GRU were implemented. However, Bi-LSTM was also implemented to demonstrate its ability to outperform all the former three architectures. The two hidden layers of opposite directions are connected to the same output. Bi-LSTM are able to understand context better due to its ability to approach a unit from both the directions. The data were divided for training, development and testing. The total sentences in the corpus were divided into 1341 as training instances, 191 as development instances and 383 as testing instances respectively. Four deep learning-based model were trained and tested namely: Simple RNN, LSTM, Bi-LSTM and GRU.

The results of Nepali POS tagging comprising of loss value calculated using cross-entropy and accuracy with simple RNN, unidirectional LSTM, and Bi-LSTM is shown in Table 7.

Table 7 Accuracy and Loss value of RNN variants

Model	Accuracy %	Loss value
Simple RNN	96.84	0.0221
Unidirectional LSTM	96.48	0.0612
Bidirectional LSTM	97.27	0.0190
GRU	96.86	0.0195

CONCLUSION

POS tagging of Nepali Text was carried out using simple RNN, LSTM, GRU and Bi-directional LSTM in a Nepali tagged corpus of tag size 40. The data set was divided into three sections i.e. training, development and testing. The accuracy obtained for simple RNN, LSTM, GRU and Bi-directional LSTM was 96.84%, 96.48%, 96.86% and 97.27% respectively.

Although some experiments have been carried out regarding POS tagging by previous researchers, the accuracy obtained by them is less than the above generated accuracy.

FUTURE WORK

In future, the vocabulary size and tag set can be increased to increase the efficiency. Similarly, reinforcement learning can be added for efficient training.

REFERENCES

Ahmad, A.Z., Rudy, H. & Mustika, I.W. (2017). A Comparison of Different Part-of-Speech Tagging Technique for Text in Bahasa Indonesia .*7th International Annual Engineering Seminar (InAES), Yogyakarta, Indonesia.*

Rana, F., Mehrnoush, S. & Pouyan, M. (2010). An Efficient Meta Heuristic Algorithm for POS-Tagging. *International Conference on Computing in the Global Information Technology (ICCGI*), IEEE.

Perez-Ortiz, J.A. & Forcada, M.L. (2001). Part-of-speech tagging with recurrent neural networks. *International Joint Conference on Neural Networks Proceedings*, IEEE.

Antony, P.J. & Soman, K.P. (2010). Kernel Based Part Of Speech Tagger For Kannada. *International Conference on Machine Learning and Cybernetics,* IEEE.

Fahim, H.M. (2006). Comparison of Different Pos Tagging Techniques For Some South Asian Languages**.** *A Thesis Submitted to the Department of Computer Science and Engineering of BRAC University*.

Yuan, T. & David, L. (2015). A Comparative Study on the Effectiveness of Part-of-Speech Tagging Techniques on Bug Reports. *SANER 2015, Montréal, Canada,* IEEE.

Firoj, A. & Shammur, A.C. (2016). Bidirectional LSTMs - CRFs Networks for Bangla POS Tagging. *19th International Conference on Computer and Information Technology, North South University, Dhaka, Bangladesh,* IEEE.

Archit, Y. (2018). ANN Based POS Tagging For Nepali Text. *International Journal on Natural Language Computing (IJNLC). **7**, No.3*.

Wang, T.L., Tiago, L., Marujo, L., Ram´on, A.F., Silvio, A., Chris, D., Alan, B. & Isabel, T. (2015). Finding Function in Form: Compositional Character Models for Open Vocabulary Word Representation. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 1520–1530, Lisbon, Portugal, 17-21 September 2015. c 2015 Association for Computational Linguistics,* ACL.

Peilu, W., Yao, Q., Frank, S.K., Lei, H. & Hai, Z. (2015). Part-of-Speech Tagging with Bidirectional Long Short-Term Memory Recurrent Neural Network. arXiv:1510.06168v1 [cs.CL].

Yushi, Y. & Zheng, H. (2016). Bi-directional LSTM Recurrent Neural Network for Chinese Word Segmentation. *Neural Information Processing: 23rd International Conference, ICONIP 2016, Kyoto, Japan, October 16–21, 2016, Proceedings, Part IV (pp.345-353)*.

Greeshma, P., Jyothsna, P.V., Shahina, K.K., Premjith, B. & Soman, K.P. (2018). A Deep Learning Approach for Part-of-Speech Tagging in Nepali Language. *International Conference on Advances in Computing, Communications and Informatics,* IEEE.