Journal of Science and Technology
ISSN: 2362-1443 (Print) 3059-9490 (Online) 2026, Vol. 5, No. 1

Enhancement of Fine-Grained Part-of-Speech Tagging for Nepali Text using BiLSTM-CRF and Word2Vec

Tika Ram Khojwar
Nepal College of Information Technology
Pokhara University, Nepal
tikaram.202828@ncit.edu.np

Pradip Paudyal*
Nepal Telecommunications Authority
Kathmandu, Nepal
ppaudyal1@gmail.com

*Corresponding author

***Abstract*— Part-of-speech (POS) tagging is an essential and foundational task in numerous Natural Language Processing (NLP) applications, including machine translation, sentiment analysis, text-to-speech conversion, speech recognition, text summarization, question answering, information retrieval, word sense disambiguation, and Named Entity Recognition. POS tagging entails assigning the correct tag to each token in the corpus, considering its context and the language's syntax. An optimal POS tagger plays a crucial role in computational linguistics. Its importance cannot be emphasized enough because inaccuracies in tagging can greatly affect the performance of complex natural language processing systems. In this work, a deep learning algorithm, BiLSTM with CRF, has been implemented for fine-grained Nepali POS tagging. The Gensim's Word2Vec word embedding has been trained on sentences and has been used as the embedding layer while creating the model. Additionally, BiLSTM-CRF with Word2Vec has been compared to the well-known models GRU, LSTM, and BiLSTM. BiLSTM-CRF with the Word2Vec word embedding performed the best and achieved a new state of the art F1 score of 99.81% for fine grained Nepali text POS tagging on the Nepali Monolingual Written Corpus.**

Keywords*— *POS Tagging, Nepali Text, Natural Language Processing, GRU, LSTM, BiLSTM, CRF.

I. Introduction

Natural Language Processing (NLP) is a subfield of artificial intelligence that focuses on enabling interaction among computers and spoken language. It encompasses the creation of algorithms and methodologies that empower computers to comprehend, interpret, and produce human language in a meaningful and practical manner. Part-of-Speech (POS) tagging constitutes a fundamental task within NLP. It involves assigning POS tags (Noun, pronoun, adjective, verb, adverb, preposition etc.) to each phrase in a sentence of a natural language. The input for the algorithm consists of a sequence of words in a natural language phrase and a predefined set of POS tags. The output is the most suitable POS tag assigned for each word in the sentence. POS tagging provides valuable information about a word and its neighboring words, which proves beneficial for various advanced NLP tasks like speech and natural language processing applications, semantic analysis, machine translation, text-to-speech conversion, text summarization, speech recognition, question answering, information retrieval, word sense disambiguation, Named entity recognition and more [3, 8, 10].

The Nepali language is characterized by its morphological richness, with a notable feature being its intricate inflectional system, especially within its verbal domain. In Nepali, a single verb can exhibit over 20 distinct inflectional forms, each encoding a range of morphological features including gender, aspect, number, mood, honorifics, tense, and person [1].

II. Literature Review

Several researches have been done in the field of POS tagging for Nepali language. Some of them used statistical model (HMM) for identifying the tags while some used supervised machine learning model and some used supervised deep learning model to train the model.

Tej Bahadur Shahi et al. [8] focused on SVM based POS tagger for Nepali language which used the dictionary as a primary resources. This dictionary was collected from the Final Nepali Corpus which contained only 11147 unique words. The POS tagging approaches like rule-based and HMM cannot handle many features that would generally be required for modeling a morphologically rich language like Nepali. SVM was efficient, portable, scalable and trainable. So, this paper proposes a SVM based tagger. The SVM tagger constructed feature vectors for each word in the input and classifies them into one of two classes using a One Vs Rest approach. The SVM algorithm achieved an accuracy rate of 96.48% for known words, 90.06% for unknown words and 93.27% in overall. That means SVM tagger demonstrated strong performance for known words. In comparison to rule-based and Hidden Markov Model approaches, the SVM-based tagger exhibits a slightly higher overall accuracy.

Sarbin Sayami et al. [5] addressed the implementation and comparison between multiple deep learning based POS taggers for Nepali text. The examined approaches include GRU, LSTM, BiLSTM and mBERT. These models were trained and assessed using Nepali English parallel corpus labeled with 43 POS tag and including almost 88000 words which were collected from Madan Puraskar Pustakalaya. The design of that Nepali POS Tagset was influenced by the

DOI : https://doi.org/10.3126/jost.v5i1.93048

Article History: Received: 7 Nov. 2025, Revised: 15 Jan. 2026, Accepted: 28 Feb. 2026.

PENN Treebank POS Tagset. The dataset was separated into three sections: training, devel opment and testing. The accuracy of basic GRU, RNN, LSTM and BiLSTM was 96.86%, 96.84%, 96.48% and 97.27% respectively. Consequently, the BiLSTM exhibited superior performance compared to the other three RNN variants.

Ashish Pradhan et al. [3] presented a comprehensive study and comparing two techniques, HMM and GRNN, for POS Tagging in Nepali text. The POS taggers aimed to address the issue of ambiguity in Nepali text. Evaluation of the taggers was performed using corpora from TDIL (Technology Development for Indian Languages) which contained a total of 424716 tagged words with 39 tags, tags followed the guidelines of ILCI (Indian Languages Corpora Initiative), BSI (Bureau of Indian Standard), with implementation carried out using Python and Java programming languages, along with the NLTK Toolkit library. The achieved accuracy rates were as follows: 100% for known words (without ambiguity), 58.29% for ambiguous words (HMM), 60.45% for ambiguous words (GRNN), and 85.36% for non-ambiguous unknown words (GRNN). Although the GRNN tagger achieved the ac curacy as high as the HMM Tagger, it failed or became unstable when the training dataset was greater than 7000 words, while the HMM Tagger was trained with more than 400000 words with corresponding tags. A total of 4000 words were used for testing on both HMM and GRNN taggers.

I. Shrestha, S. S. Dhakal [1] applied three deep learning models: BiGRU, BiLSTM, and BiLSTM-CRF for fine grain POS tagging in the Nepali text. It used Nepali National Corpus (NNC). It has 17 million semi-manually and manually tagged words with 112 POS tags. Their findings revealed that deep learning models effectively capture fine grained morphological attributes such as number, gender, person, and honorifics inherent in incredibly inflectional languages such as Nepali, given a substantial dataset. The research experimented with two types of embeddings: randomly initialized Bare embedding and pre-trained multilingual BERT. Surprisingly, training with randomly initialized Bare embedding outperformed models trained with massive pre-trained multi-lingual BERT embedding on Nepali downstream tasks such as POS tagging. Among the models tested, BiLSTM-CRF utilizing the Bare embedding achieved the highest performance, setting a new state of the art F1 score of 98.51% for fine grained Nepali POS tagging. This study contributes significantly to the advancement of NLP techniques tailored specifically for the Nepali language

Despite the extensive research on POS tagging for the Nepali text using various deep learning models such as BiLSTM, GRU, and LSTM, and traditional techniques like HMM and SVM, there exists a noticeable research gap that demands attention. The current literature predominantly focuses on addressing the challenges posed by the morphologically rich and complex nature of Nepali, with a primary emphasis on achieving high accuracy rates. However, the persistently high false positive rates in existing POS tagging models for Nepali present a significant challenge. Most studies utilize deep learning algorithms with coarse-grained tagsets, and while some explore fine-grained tagsets, the issue of false positives remains inadequately addressed. Furthermore, there is a lack of comprehensive comparative studies evaluating the effectiveness of these models, particularly in the context of reducing false positives. This research gap underscores the need for a more nuanced approach, considering the unique linguistic features of Nepali, and a more in-depth exploration of fine-grained tagsets. The proposed study aims to fill this gap by introducing a BiLSTM-CRF model with Word2Vec embeddings on fine-grained tagsets, specifically designed to mitigate the persistent challenge of false positives in Nepali POS tagging. Additionally, it seeks to provide a comparative analysis with existing models such as GRU, LSTM, and BiLSTM, shedding light on the relative effectiveness of the proposed model in addressing the identified issues.

III. Methodology

In this chapter, we present the proposed method for determining the POS tags of the provided text. The following figure provides an overview of the tagging process.

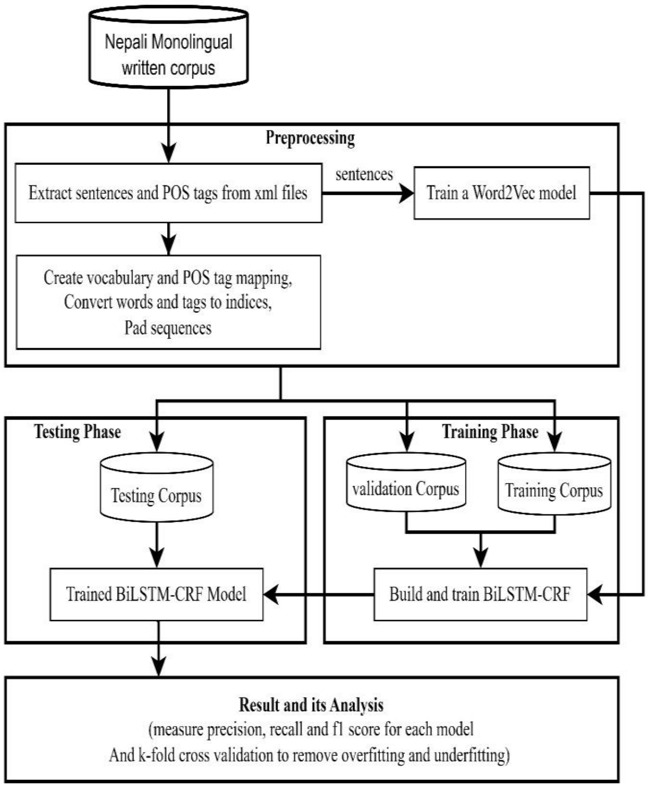

Fig. 1. Proposed methodology for POS tagging

A. *Data Collection*

This research study has implemented morphologically annotated Nepali National Corpus (NNC). NNC is collection of xml files, has over 17 million words with 112 Nelralec tagset. Each xml files are in same format, sentences wrap within “s” element with “n” attribute and words are wrap within “w” element with “ctag” attribute. Ctag represents POS tag of crossponding word. The corpus is divided into two main parts: the core corpus (core sample) and the general corpus. The core sample (CS) is a compilation of Nepali written texts from 15 diverse genres, with each text containing 2000 words. These texts were published between 1990 and 1992. On the other hand, the general corpus (GC) comprises written texts gathered from various sources, including the internet, newspapers, books, publishers, and authors.

B. *Pre-processing*

1) *Extract Sentences and POS Tags from XML files:*

Nepali monolingual written corpus files are in XML format. We cannot use XML files as input to train a deep learning model. Therefore, we first extract sentences and their corresponding parts-of-speech tags in list form. For example:

all_sentences = [['सत्तापक्ष', 'सित', 'नजीक', 'रहेर', 'काम', 'गर्न', 'विपक्षीदल', 'प्रतिवद्ध'], ['काठमाडौं', ',', 'असोज', '१६', '।'], …]

all_pos_tags = [['NN', 'II', 'RR', 'VQ', 'NN', 'VI', 'NN', 'JX'], ['NP', 'YM', 'NN', 'MM', 'YF'], ...]

2) Train Word2Vec word embedding on sentences:

Word2Vec is a popular word embedding technique in the fields of NLP and ML. It is used to represent words as dense vectors in a continuous vector space, with words with similar meanings clustered together. Word2Vec models are trained on large text corpora and are capable of capturing semantic relationships between words. To achieve this, we first created a Word2Vec model using Gensim in Python. We then passed all sentences as an argument to the created model, resulting in the embedding model.

3) Create Vocabulary and POS Tag Mapping:

Vocabulary and POS tag mappings convert words and tags into numerical indices for deep learning models. For vocabulary and POS tag mappings, we firstly create vocab set and pos_tags_set set for unique words and unique POS tags respectively. Then create word_to_idx dictionary to maps each unique word in the vocabulary to a unique index. And same for the tag_to_idx. Then create idx_to_tag dictionary. It is reverse of the tag_to_idx dictionary. We also use 'UNK' token to handle out-of vocabulary words or tags that are not present in the initial training data. Following are sample of the word_to_idx and tag_to_idx dictionaries.

word_to_idx = {'समर्थित': 1, 'व्यक्तिविशेष': 2, 'खुलेछन्': 3, 'चर्कँदो': 4, …}

tag_to_idx = {'IE': 1, 'VVTN1': 2, 'DGF': 3, 'PTH': 4, 'YQ': 5, …}

4) Convert Words and Tags to Indices:

It is the process where we converted words and POS tags to their corresponding numerical indices using the mappings created earlier (word_to_idx and tag_to_idx). Each inner list represents a sentence, and it contains numerical indices corresponding to the words or POS tags in that sentence. Following are the sample of sentences_indices and pos_tags_indices list of lists.

sentences_indices = [[24283, 33809, 29171, 17993, 31405, 11612, 27897, 29351], [39860, 14035, 16097, 37307, 10206], …]

pos_tags_indices = [[20, 89, 25, 24, 20, 43, 20, 88], [44, 39, 20, 55, 46], …]

5) Pad Sequences:

A Padding sequences refers to the process of making all input sequences in a dataset of the same length by adding special tokens (usually with a value of 0) to the shorter sequences. This is typically done to facilitate the training of neural networks, as they usually expect inputs of uniform length. Following are the sample of padded_sentences_indices and padded_pos_tags_indices list of lists. Let, length of longest sentence is 10.

padded_sentences_indices = [[0, 0, 24283, 33809, 29171, 17993, 31405, 11612, 27897, 29351], [0, 0, 0, 0, 0, 39860, 14035, 16097, 37307, 10206], …]

padded_pos_tags_indices = [[0, 0, 20, 89, 25, 24, 20, 43, 20, 88], [0, 0, 0, 0, 0, 44, 39, 20, 55, 46], …]

C. Algorithms

1) GRU:

A GRU is a type of RNN architecture that have shown effectiveness in various NLP tasks, including POS tagging. GRU has two gates, an update gate and a reset gate, which control information flow in and out of the hidden state [5]. The update gate helps retain relevant information from the past while the reset gate helps update the hidden state with new information. GRUs can result in fewer parameters, making them computationally more efficient and easier to train, especially in scenarios with limited data. GRUs share parameters across different positions in a sequence which allows the model to generalize patterns learned at one position to other positions, which is beneficial for tasks like POS tagging where similar linguistic patterns may occur at different positions in a sentence. Pre-trained word embedding can be used as input to a GRU model for POS tagging, leveraging transfer learning. The GRU can then adapt to the specific POS tagging task, benefiting from the semantic relationships captured by pre-trained embedding. GRUs have demonstrated competitive performance in various NLP tasks, and they are considered a state-of-the-art choice for certain applications, especially in cases where computational efficiency is crucial.

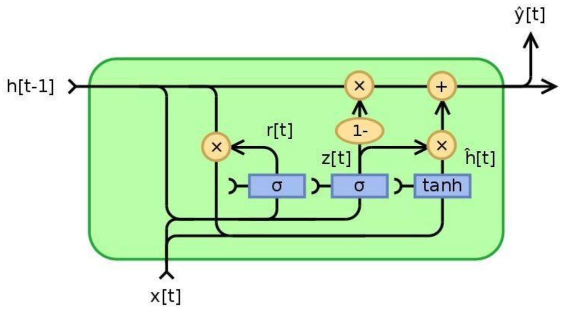

Fig. 2. GRU model [13]

2) LSTM:

LSTM is a type of RNN architecture that have been widely used in NLP tasks, including POS tagging. LSTMs are designed for sequential data processing, which aligns well with the nature of POS tagging. In POS tagging, the order of words in a sentence is crucial for understanding the syntactic structure. LSTMs also have a memory cell that can retain information over long sequences, allowing the model to capture dependencies between words that are separated by significant distances within a sentence [10]. This is crucial for accurately assigning POS tags. Pre-trained word embedding can be used as input to an LSTM model for POS tagging, leveraging transfer learning. The LSTM can then adapt to the specific POS tagging task, benefiting from the semantic relationships captured by pre-trained embedding. LSTM can also handle variable-length sequences, making them suitable for POS tagging tasks where sentences have different lengths. LSTM have demonstrated state-of-the-art performance in various NLP tasks, including POS tagging. They have been widely adopted and have shown effectiveness in capturing complex linguistic patterns.

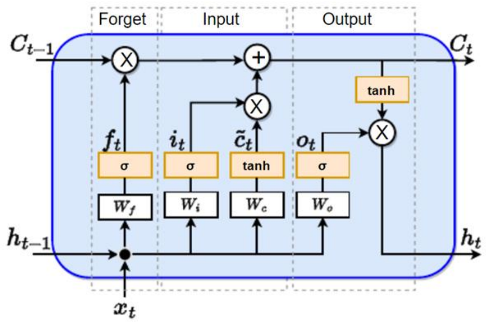

Fig. 3. LSTM cell with gating controls [14]

3) *BiLSTM:*

BiLSTM is well-suited for POS tagging tasks due to their ability to capture contextual information from both the past and the future. BiLSTM can capture long-term dependencies in sequential data [5]. The bidirectional nature of BiLSTM helps in understanding the syntactic and semantic context in a more holistic way [10]. This is particularly important in POS tagging, where the correct assignment of a POS tag often relies on the broader linguistic context. POS tagging can be challenging when words have multiple possible POS tags depending on the context. BiLSTM, by considering both past and future context, are better equipped to handle such ambiguity and make more informed predictions. BiLSTM can handle variable-length sequences, making them suitable for POS tagging tasks where sentences have different lengths. BiLSTMs have demonstrated state-of-the-art performance in various NLP tasks, including POS tagging [1]. They are widely used in research and industry due to their effectiveness in capturing complex linguistic patterns.

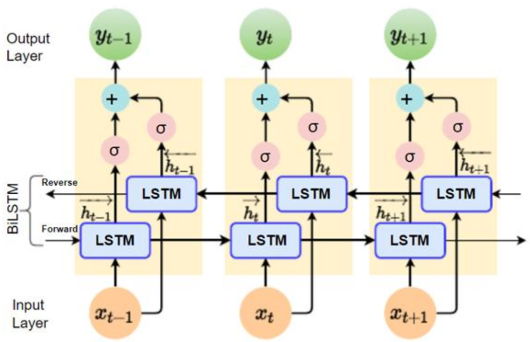

Fig. 4. Architecture of the BiLSTM network [14]

4) *BiLSTM-CRF:*

BiLSTM-CRF is commonly used for POS tagging due to its capacity to capture bidirectional contextual information, making it effective in modeling long-range dependencies within a sequence. The combination of BiLSTM layers and a CRF layer addresses the need for accurate POS tagging, where the context of a word within a sentence and dependencies with neighboring words play a crucial role. The bidirectional processing allows the model to consider both preceding and succeeding words, while the CRF layer explicitly models label dependencies, contributing to the architecture's success in sequence labeling tasks. Its consistent state-of-the-art results on POS tagging benchmarks underscore its effectiveness in handling the complexities of natural language processing tasks requiring nuanced understanding of contextual relationships and sequential dependencies.

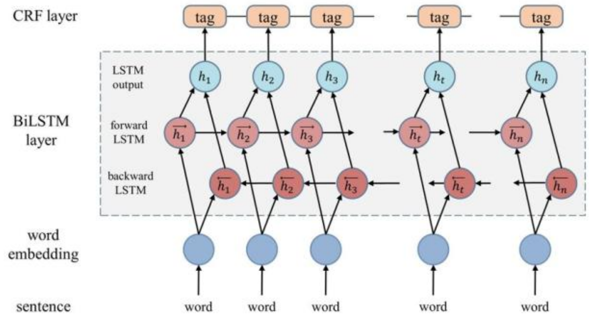

Fig. 5. BiLSTM-CRF structure [15]

IV. Results and Analysis

This chapter deals with analyzing the results of the BiLSTM-CRF with Word2Vec and established models like GRU, LSTM, and BiLSTM. For all four models, the dataset has been split into a 70:15:15 ratio for training, validation and testing, respectively.

A. *Experiment using BiLSTM-CRF with Word2Vec*

```
Layer (type)                 Output Shape              Param #
=================================================================
input_3 (InputLayer)         [(None, 303)]             0

embedding_2 (Embedding)      (None, 303, 100)          4043400

bidirectional_2 (Bidirecti   (None, 303, 200)          160800
onal)

time_distributed_2 (TimeDi   (None, 303, 104)          20904
stributed)

dropout_2 (Dropout)          (None, 303, 104)          0

crf_layer (CRF)              [(None, 303),             21630
                              (None, 303, 103),
                              (None,),
                              (103, 103)]

activation_2 (Activation)    (None, 303, 103)          0

=================================================================
Total params: 4246734 (16.20 MB)
Trainable params: 4246734 (16.20 MB)
Non-trainable params: 0 (0.00 Byte)
```

Fig. 6. Summary of BiLSTM-CRF with Word2Vec

BiLSTM-CRF with Word2Vec model with 10 epochs, 128 batch size, 0.3 dropout, Adam optimizer, 0.001 learning rate, sparse_categorical_crossentropy loss function gave 99.81 % F1-score and 99.82 % accuracy.

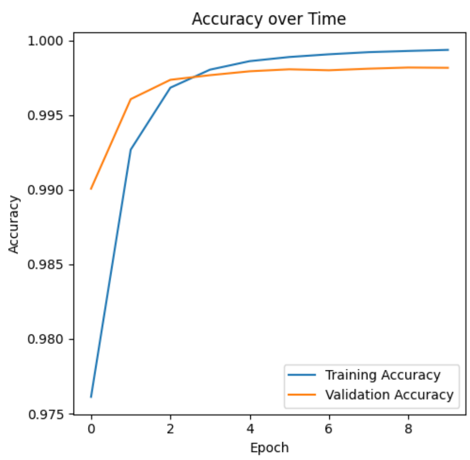

Fig. 7. (a) Accuracy over time of BiLSTM-CRF with Word2Vec model

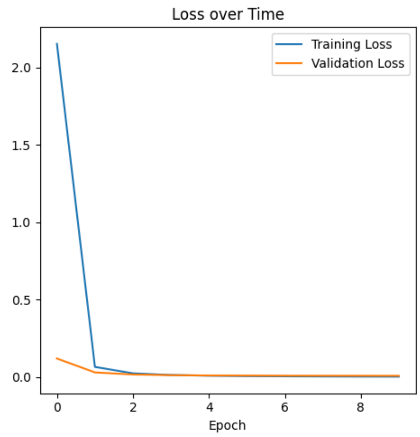

Fig. 1(b) Loss over time of BiLSTM-CRF with Word2Vec model

Figure 7 shows that the training and validation accuracy are increasing over epochs. Training accuracy is higher than the validation accuracy. After the 3rd and 4th epochs validation accuracy and training accuracy increases slowly respectively. Training and validation loss are decreasing over epochs. That means it minimizes the error in the model.

```
Precision: 0.9982
Recall: 0.9982
F1 Score: 0.9981
Accuracy: 0.9982
```

Fig. 8. Precision, Recall, F1 score and Accuracy

Figure 8 shows that Precision, Recall, F1 Score and Accuracy of BiLSTM-CRF model, all are above 99%.

```
Enter a sentence: सशस्त्र प्रहरी सीमा सुरक्षार्थ अग्र भाग मा तैनाथ सुरक्षा निकाय हो । तर, अहिले सम्म सीमा सुरक्षा ऐन नै आउन सके को छैन ।
1/1 [==============================] - 0s 125ms/step
सशस्त्र: JX
प्रहरी: NN
सीमा: NN
सुरक्षार्थ: NN
अग्र: NN
भाग: NN
मा: II
तैनाथ: JX
सुरक्षा: NN
निकाय: NN
हो: VVYN1
।: YF
तर: CC
,: YM
अहिले: RD
सम्म: II
सीमा: NN
सुरक्षा: NN
ऐन: NN
नै: TT
आउन: VI
सके: VE
को: IKM
छैन: VVYN1
।: YF
```

Fig. 9. Precision, Recall, F1 score and Accuracy

Figure 9 shows that while testing the BiLSTM-CRF model with user input, the model has been able to capture fine grained morphological properties such as number, gender, person, and honorifics.

B. Comparing Results between GRU, LSTM, BiLSTM and BiLSTM-CRF with Word2Vec

Fig. 10. Comparison between BiLSTM-CRF with Word2Vec and established models GRU, LSTM, and BiLSTM based on Train_Accuracy, Val_Accuracy and Test Accuracy

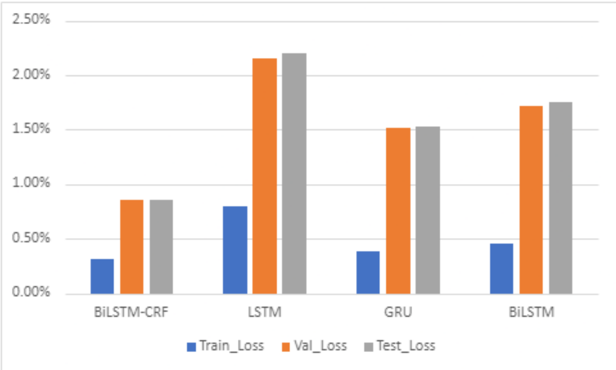

Fig. 11. Comparison between BiLSTM-CRF with Word2Vec and established models GRU, LSTM, and BiLSTM based on Train_Loss, Val_Loss and Test_Loss

TABLE I. COMPARISON BETWEEN BILSTM-CRF WITH WORD2VEC AND ESTABLISHED MODELS GRU, LSTM, AND BILSTM BASED ON TRAIN_LOSS, TRAIN_ACCURACY, VAL_LOSS, VAL_ACCURACY, TEST_LOSS AND TEST_ACCURACY

Models	Train _Loss	Train _Acc	Val_ Loss	Val _Acc	Test _Loss	Test _Acc
BiLSTM-CRF with Word2Vec	**0.0031**	**0.9994**	**0.0086**	**0.9982**	**0.0086**	**0.9981**
LSTM	0.0080	0.9981	0.0215	0.9966	0.0220	0.9964
GRU	0.0039	0.9990	0.0152	0.9971	0.0153	0.9969
BiLSTM	0.0046	0.9991	0.0172	0.9977	0.0175	0.9976

Table I presents a comparative analysis of various models trained on a dataset, showcasing their performance metrics across training, validation, and test sets. Four models are evaluated: BiLSTM-CRF with Word2Vec embedding, LSTM, GRU, and BiLSTM. In terms of training loss, BiLSTM-CRF with Word2Vec achieved the lowest value of 0.0031, indicating better fitting to the training data compared to the other models. Similarly, it attained the highest training accuracy of 0.9994, suggesting superior learning capability. However, when considering validation and test sets, its performance slightly degraded, with a validation loss of 0.0086 and a test loss of 0.0086, accompanied by validation and test accuracies of 0.9982 and 0.9981, respectively. Among the other models, BiLSTM and GRU also demonstrate competitive performances, with relatively lower training losses and high accuracies. LSTM, on the other hand, exhibits slightly higher losses across all datasets, indicating comparatively poorer performance.

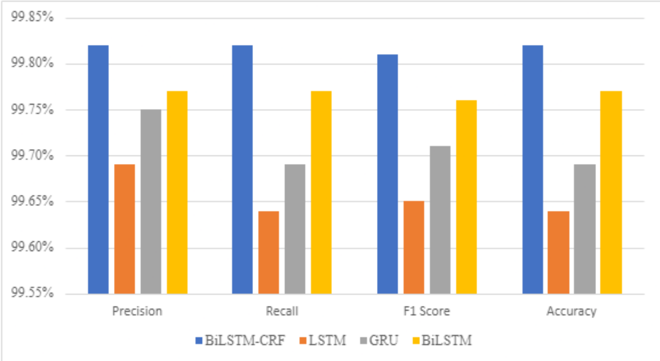

Fig. 12. Comparison between BiLSTM-CRF with Word2Vec and established models GRU, LSTM, and BiLSTM based on precision, recall, f1 score and accuracy

TABLE II. COMPARISON BETWEEN BILSTM-CRF WITH WORD2VEC AND ESTABLISHED MODELS GRU, LSTM, AND BILSTM BASED ON PRECISION, RECALL, F1 SCORE AND ACCURACY.

Models	Precision	Recall	F1 Score	Accuracy
BiLSTM-CRF with Word2Vec	**0.0031**	**0.9994**	**0.0086**	**0.9982**
LSTM	0.0080	0.9981	0.0215	0.9966
GRU	0.0039	0.9990	0.0152	0.9971
BiLSTM	0.0046	0.9991	0.0172	0.9977

Table II shows that the "BiLSTM-CRF with Word2Vec" model performs the best across all metrics, with high precision, recall, F1 score, and accuracy. The BiLSTM and GRU models also perform well, though slightly lower than the BiLSTM-CRF with Word2Vec. The LSTM model demonstrates the lowest performance among the models listed in terms of precision, recall, F1 score, and accuracy.

V. CONCLUSIONS AND FUTURE WORKS

In this work, we applied BiLSTM-CRF with Word2Vec model to achieve fine-grain POS tagging in Nepali. This study showed that hybrid model BiLSTM-CRF with Word2Vec captured fine grained morphological properties such as person, number, gender, and honorifics embedded within words in largely inflectional languages such as Nepali. This study also able to improve accuracy by reducing false positive rate in fine grain POS tagging in Nepali text. The POS-tagged Nepali National Corpus, which contains over 17 million words, and the NELRALEC tag sets, which include 112 fine-grained POS tags was used for modelling the POS tagger. Additionally, we assessed how well the BiLSTM-CRF performed with Word2Vec in comparison to the well-known models GRU, LSTM, and BiLSTM. BiLSTM-CRF with Word2Vec achieved better result compare with the state of the art models with F1 score of 99.81% for fine grained Nepali POS tagging.

Although our models have demonstrated remarkable results in identifying fine-grained tags, there is still much work to be done in the future.

a. Utilize a balanced and up-to-date systematic dataset to make more robust Nepali POS tagger.
b. Address ambiguous words that possess different contextual meanings within a sentence.
c. Develop strategies to handle unknown words.
d. To test and validate the model with bigger dataset for better learning.

REFERENCES

[1] I. Shrestha and S. S. Dhakal, "Fine-grained part-of-speech tagging in Nepali text," in *Procedia Computer Science*, vol. 189, pp. 300-311, 2021. DOI: 10.1016/j.procs.2021.05.099.

[2] Chiche and Yitagesu, "Part of speech tagging: a systematic review of deep learning and machine learning approaches," *Journal of Big Data*, vol. 9, no. 10, 2022, https://doi.org/10.1186/s40537-022-00561-y.

[3] A. Pradhan and A. Yajnik, "Probabilistic and Neural Network Based POS Tagging of Ambiguous Nepali Text: A Comparative Study," in *ISEEIE 2021: 2021 International Symposium on Electrical, Electronics and Information Engineering*, pp. 249–253, Feb. 2021.

[4] A. Yajnik, "Part Of Speech Tagging Using Statistical Approach for Nepali Text," *International Scholarly and Scientific Research & Innovation,* vol. 11, no. 1, 2017.

[5] S. Sayami, T. B. Shahi, and S. Shakya, "Nepali POS Tagging Using Deep Learning Approaches," *NU. International Journal of Science*, Dec. 2019.

[6] A. Paul, B. S. Purkayastha, and S. Sakar, "Hidden Markov Model Based Part of Speech Tagging for Nepali Language," in *International Symposium on Advanced Computing and Communication (ISACC)*, Sep. 2015.

[7] A. Yajnik, "General Regression Neural Network Based POS Tagging for Nepali Text," in *4th International Conference on Natural Language*, pp. 35–40, 2018.

[8] *T. B. Shahi, T. N. Dhamala, and B. Balami, "Support Vector Machines Based Part Of Speech Tagging for Nepali Text," International Journal of Computer Applications (0975 – 8887), vol. 70, no. 24, May 2013.*

[9] A. Yajnik, "ANN Based POS Tagging for Nepali Text," *International Journal on Natural Language Computing (IJNLC)*, vol. 7, no. 3, June 2018.

[10] G. Prabha, P. V. Jyothsna, K. K. Shahina, B. Premjith, and K. P. Soman, "A Deep Learning Approach for Part-Of-Speech Tagging in Nepali Language," in *International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Sep. 2018.

[11] M. Jayaweera and N. G. J Dias, "Hidden Markov Model Based Part of Speech Tagger for Sinhala Language," *International Journal on Natural Language Computing (IJNLC)*, vol. 3, no. 3, June 2014.

[12] P. Sinha, N. M. Veyie, and B. S. Purkayastha, "Enhancing The Performance of Part Of Speech Tagging of Nepali Language Through Hybrid Approach," *International Journal of Emerging Technology and Advanced Engineering*, vol. 5, no. 5, May 2015.

[13] ["Gated Recurrent Unit," commons.wikimedia.org. Online]. Available: https://commons.wikimedia.org/wiki/File:Gated_Recurrent_Unit,_type_1.svg. [Accessed: Sep. 23, 2023].

[14] B. Bohara, R. I. Fernandez, V. Gollapudi, and X. Li, "Short-Term Aggregated Residential Load Forecasting using BiLSTM and CNN-BiLSTM," in *International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT)*, 2022.

[15] S. Park, X. Wang, C. C. Menassa, V. R. Kamat, and J. Y. Chai, "Natural Language Instructions for Intuitive Human Interaction with Robotic Assistants in Field Construction Work," arXiv preprint arXiv:2307.04195, 2023. [Online]. Available: https://arxiv.org/abs/2307.04195

[16] M. Maimaiti, A. Wumaier, K. Abiderexiti, and T. Yibulayin, "Bidirectional Long Short-Term Memory Network with a Conditional Random Field Layer for Uyghur Part-Of-Speech Tagging," *Information*, vol. 8, no. 4, p. 157, Nov. 2017. [Online]. Available: https://doi.org/10.3390/info8040157.

[17] Y. Xiong et al., "A fine-grained Chinese word segmentation and part-of-speech tagging corpus for clinical text," in *BMC Medical Informatics and Decision Making*, vol. 19, no. Suppl 2, p. 66, Dec. 2019. [Online]. Available: https://doi.org/10.1186/s12911-019-0770-7.

[18] S. Kumar, M. Anand Kumar, and K.P. Soman, "Deep Learning Based Part-of-Speech Tagging for Malayalam Twitter Data," *J. Intell. Syst.*, vol. 28, no. 3, pp. 423–435, 2019. DOI: 10.1515/jisys-2017-0520.

[19] F. Zaman, O. Maqbool, and J. Kanwal, "Leveraging Bidirectional LSTM with CRFs for Pashto tagging", in *IEEE Transactions on Asian Low-Resource Language Information Processing*, vol. 9, no. 4, pp. 39-52, Sept. 2018.

[20] Z. Huang, W. Xu, and K. Yu, "Bidirectional LSTM-CRF Models for Sequence Tagging," *Baidu Research*, Aug. 2015. [Online]. Available: arXiv:1508.01991v1 [cs.CL].

[21] W. K. Birhanie and M. Butt, "Automatic Amharic Part of Speech Tagging (AAPOST): A Comparative Approach Using Bidirectional LSTM and Conditional Random Fields (CRF) Methods," in *Advances of Science and Technology: 7th EAI International Conference, ICAST 2019, Proceedings*, N. G. Habtu, D. W. Ayele, S. W. Fanta, et al. (Eds.), Cham, Springer, 2020, pp. 512-521. [Online]. Available: http://nbn-resolving.de/urn:nbn:de:bsz:352-2-13uilhq9epned7.