International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(1s) 2026 pp. 1-15

https://doi.org/10.70917/ijcisim-2026-1995
Article

Part- Of-Speech (Pos) Tagging of Limbu Language Using Artificial Neural Networks

Abigail Rai[1]*, Samarjeet Borah[2]

[1] Department of Computer Applications, Sikkim Manipal Institute of Technology, Majhitar, Sikkim Manipal University, Gangtok, Sikkim, India.angellaraiz90@gmail.com
[2] Department of Computer Applications, Sikkim Manipal Institute of Technology, Majhitar, Sikkim Manipal University, Gangtok, Sikkim, India. samarjeet.b@smit.smu.edu.in

Abstract: This paper aims to explore the use of neural networks for part-of-speech tagging (PoS) of Limbu language, a Tibeto-Burman language spoken in eastern Nepal, North-Eastern India, Bhutan etc. The Limbu language has a complex morphology with a rich system of inflections and derivations. Computational implementation for an under-resource language is strenuous for any natural language processing works. All through the process of part-of-speech tagging of Limbu language we face many impediments as our work is the initial work or the only towards Natural language Processing of Limbu language. There exists no computational work for the language as well all the pre-processing needed to be carried out by us including corpus generation. In this study we used a typical approach Feedforward Neural Network. The performance of the tagger is evaluated using various metrices such as accuracy, precision, recall and F1 score. We also compared the achievement of implemented model with other stochastic models like Hidden Markov Model (HMM). Our findings show that the neural network-based tagger achieves competitive results, outperforming stochastic models, and has the potential to be used in various natural language processing applications for Limbu language. The applied method imparted excellent accuracy for the corpus used. INDEX TERMS Limbu, Part-of-Speech (PoS) Tagging, Natural Language Processing (NLP), Hidden Markov Model (HMM), Feed Forward Neural Network, BiLSTM.
Keywords: NA

1. Introduction

Use of text processing has become an essential step in this age of modern technology. Translating between humans and robots is a two-way street, so natural languages knowledge is required. Powerful and Cost-Effective natural language processing with innovative technologies to process more com- plex languages. By analyzing texts and utterances, utilizing a variety of language structures it enriches the logical part in rational human-machine interactions. PoS-tagging is one the main and most important elements to work with NLP (Natural Language Processing) fields. It is a key component in natural language processing (NLP) by contextualizing each morpheme to its position within classes of lexical components known as parts-of-speech.

One after another, the rapid development of communication and information technology has led to a great interest in NLP technologies. As a result, numerous technologies and NLP techniques are being developed for many languages. Many more languages remain endangered. Though numerous languages such as Limbu have seen little in the way of computational work, it is important to make these machine-readable for their conservation The piece of PoS-tagging (Natural Language Processing technique) will map onto the text using all kinds of words in a paragraph or sentence. PoS-tagging has a quite few of tools and techniques, have tried a feed- forward neural network with sharp (stochastic) methods to create empirically based PoS tags in consideration with the barrier Limbu language. Limbu is spoken by the Limbu people in eastern Nepal, India (particularly in Sikkim, Darjeeling and Assam) as well as ex- patriate communities worldwide including Bhutan, Burma, Thailand, the UK, Hong Kong, Singapore, North America, Australia during last thirty years. They speak the Yakthungpan language, identifying themselves as the Yakthung (Yakthumba). Although the majority of people speak in Limbu, it has not been computationally processed.

Our work makes a substantial contribution to the language and its related com- munity through providing essential computational support for Yakthung (Limbu).

This study describes the process of creating PoS-tagging methods for a low resource language, Limbu. Broadening our approach, we explore challenges from a general linguistic perspective: morphosyntactic feature complexity in Limbu and its low availability of annotated data compared to high-resource languages.

The key contribution of this work as follows:

- Provides the first of annotated corpora for Limbu POS tagging, paving way for computational experimentation for bloodstain language.
- This work performs a systematic comparison between traditional statistical approaches (Hidden Markov Model) and neural architectures (Feedforward Neural Network and BiLSTM) for Limbu POS tagging.
- Analysis of how morphological complexity impacts tagging performance and demonstrate through experimentation the issues raised by current-best architectures when applied to a morphologically rich low-resource language.

A. Research Motivation

Despite its cultural significance, the Limbu language, which belongs to a Limbu community in Nepal and some parts of India has been left behind in computational linguistics. Although most significant advances have been made in NLP field, there is still considerable lack of computational work on Limbu language. This research venture to fill the gap by advancing computational endeavor attune to the language Limbu.

1) Computational Gap

This work aims to bridge the wide gap in computational resources for Limbu language. At present, there is no computational work carried out for natural language processing of Limbu language. It set the foundation necessary for building a language processing system, as well as leading to hints of expected values in PoS-tagging systems. This pioneering map not only provides an initial corpus formation but also preparing in tools for future sophisticated language processing.

2) Advancing NLP for Limbu

Developments on PoS-tagging system grasp enormous importance in advancing NLP in the future progress. Information retrieval, sentiment analysis, and machine translation are just a few of the many NLP applications that depend on PoS-tagging. With the computational progression of PoS-tagger framework, this research opens doors for amalgamation of the Limbu language into advanced technological applications. This not only stimulate further research and advancement in the particular domain but as well promote the languages engagement in the wider technological landscape.

3) Empowering the Limbu community

This research carries the potential to capacitate the Limbu community, in the era of Internet of Things (IOT) and digital transformation. As the globe increasingly interconnected, computational tools for minority languages are pivotal for preservation of culture and involvement. A PoS-tagging system can work as stepping stone for creating educational tools, digital content, and applications in Limbu, consequently encouraging the community to thrive and preserving linguistic heritage in a promptly evolving digital landscape.

In this paper we look to address these domains not only dynamically in the computational linguistics field for that contributes immensely but also validate and allow the Limbu communities cultural and technological advancements. Development of PoS-tagging system for Limbu language is the first and foremost initiative to carry on with language empowerment in present day digital world.

2. Related Work

The deep learning approach [1] for part-of-speech (PoS) tagging for small size corpus has proved successful utilizing tagged corpus and rich vector representation, explored autoencoder-based approaches and bidirectional LSTM autoencoders. Penn tree bank has been successfully used to tag unknown words with cyclic redundancy network [2]. For developing PoS-tagger, training and test set development is also important.

PoS-tagging requires segmentation of errors and Vishaal Jatav et al. [3] segmented errors as critical error and non-critical errors, which helped immensely while tagging part-of-speech to texts in sentence level. Morphological analysis being the crucial phase needs immense processing, especially for the agglutinative languages like Chinese and Japanese [4]. Supervised PoS-tagging performance increases as the size of the corpus increases, which was tested with the languages English and Bangla [5].

As compared to single-neuro tagger with a fixed context, multi-neuro tagger, HMM tagger and CRF based taggers perform better. Multi neuro tagger training time is less study [6] highlights that same may provide suitable PoS-tagging. As it is discussed in [7], focusing on PoS- tags associated to answer would be helpful which might increase accuracy, even with the classification.

Less Corpus will degrade the PoS-tagger performance, as corpus increases the performance too. 80.56 [8] Nepali language was tagged along with the statistical approach and also having some modifications of their own approach, which was successful to present 93.156 % accuracy, which is more than statistical approach. Most of the PoS-taggers for Indian languages use combinations of language-independent techniques or stochastic methods and linguistic knowledges. Although taggers achieved good accuracy, they are less useful for other inflective Indian languages as their performance depends more on linguistic knowledge. They used HMM based tagger in [9] with naive (longest suffix matching) stemmer as pre-processor and obtained accuracy performance of 93.12 %. The taggers performances could be analyzed while designing the earth truth set, which includes tagged words related with sampled corpus effectives [10]. [11] These results show that for low resource languages, given proper corpus size LSTM-RNN based tagger outperforms SVM and the CRF, its rivals in [1], regardless of corpus size. For some of the complex and low-resourced languages, formulating annotation system that does not require natural definition of words is Possible [13].

CRF-based sequence models are found to be always better than neural ones in terms of F1-score, and LSTM/GRU models have Positive but not consistent improvement under resource-poor conditions [14]. Also, agglutinative morphology and transliteration variation are significant challenges, and accordingly character-level representations as well as language-aware features are essential for effective PoS-tagging. [15], this paper demonstrates that PoS-tagging in code-mixed Indian languages is difficult because of the lack of data and noise and is best addressed with sequence-based models. It concludes that language identification and context features are important to improve PoS-tagging quality. This [16] paper shows that sequence models using machine learning are also attractive for PoS-tagging in low-resource and noisy text. The experiments show that the selection of features, and context information relates a cardinal role in the quality of tagging.

With the goal of optimizing the activation function in neural networks, [17] sought to process training input as efficiently as Possible. It raises important questions about how to increase the accuracy of PoS-tagging in low-resource languages. Their novel approach [18], which combines deep learning and neural networks [19, 22], implies that low resource languages can benefit greatly from the application of neural network techniques. By increasing accuracy without significantly depending on tagged data, unsupervised method and CRF proved to be very successful in processing the language's complicated morphological structure [20]. By integrating knowledge from higher-resource languages, transfer learning dramatically increases the accuracy of language processing [21]. The ensemble technique [24], current NLP capabilities [23] and deep learning [25, 26] can improve language processing for underrepresented languages.

Table 1: Summary of Literature review on Part of Speech (POS)-Tagging

Paper	Language	Dataset	Model
Prakhar Srivastava et al. [2018]	Sanskrit	Sanskrit JNU corpus	Autoencoder based approaches, bidirectional LSTM
Kristina Toutanova et al. [2003]	English	Penn Tree-bank WSJ	HMM, CMM
Vishaal Jatav et al. [2017]	English	Penn Tree Bank 3	Hybrid Approach

Chenchen Ding et al. [2019]	Burmese (Myanmar)	Asian Language Tree-bank (ALT) Burmese Corpus	CRF, LSTM, RNN
Fahim Muhammad Hasan et al. [2007]	Bangla	Bangla Corpus	N-Gram, HMM and Brill's tagger
Ankur Parikh et al. [2009]	Hindi	ILMT	Novel Approaches, HMM, CRF
Saranlita Chotirat et al. [2021]	Thai	(TREC-6 dataset and Thai sentence dataset	CNN, BiLSTM Model, MNB, LR, SVM
Prajadhip Sinha et al. [2015]	Nepali	550 sentences, 15,720 words corpus	Hybrid (Rule based and HMM)
Manish Shrivastava et al. [2008]	Hindi	60,000 words	Trigram HMM, Na¨ıve Stemming
Swati Tyagi et al. [2016]	English	Brown Corpus	N-gram, HMM
Hour Kaing et al. [2021]	Khmer	20,000 Sentence Khmer corpus	CRF, LSTM
C.Ding et al. [2018]	Burmese, Khmer	Annotated corpus	NOVA
Jamatia, A. et al. [2015]	English and Hindi	Twitter data	CRF and LSTM/GRU
Mandal, A. et al. [2022]	Telegu and English	ICON-2015, ICON-2016, FIRE-2020 shared-task dataset, Twitter and Social media data	HMM, CRF, SVM, Naïve Bayes, MEMM, LSTM, BiLSTM, GRU and Transformer based models
Tang H. et al. [2024]	Programming Language Identifiers	IDData, MNTrain and Newman's dataset	HMM. CRF, Neural Architectures
D. Baishya et al. [2024]	Assamese	PoS annotated text	Deep Learning Architecture
S. K. Nambiar et al. [2023]	Malayalam	Parallel Corpus of Malayalam text	Neural Network Model
S. Warjri. et al. [2021]	Khasi	96,100 tokens and 6,616 words	BiLSTM-CRF and Character-based Embedding+ BiLSTM
N. Bölücü et al. [2019]	Turkish, Hungarian, Finnish and	Unlabeled Corpora of different languages	Bayesian Hidden Markov Model and Neural Model

	English		
H.Wang et al. [2019]	Singlish	Singlish Universal Dependencies Treebank	Transfer Learning with Neural Models
Y. Li et al. [2022]	Tibetan	Manually curated Tibetan Corpus	BiLSTM, IDCNN and CRF
T. Dalai et al. [2024]	Odia	PoS Tagged and Manually annotated data	Transformer Language Model
D. Pathak et al. [2023]	Assamese	PoS Annotated Corpus of ~404, 000 tokens	Ensemble Approach (Deep Learning + Rule-based Tagger)
H. Visuwalingam et al. [2021]	Tamil	Tamil PoS-Annotated Corpus and AU-KBC annotated Corpus	BiLSTM
H. Visuwalingam et al. [2024]	Tamil	AU-KBC Annotated Corpus and MeitY Corpus	Deep Learning Model

Table 1 provides a comparative appreciation on PoS-tagging, showing that different methods can be very effective to different linguistic scenarios in terms of adaptability even regarding low-resource languages.

3. Dataset Analysis

Because there was no corpus available which has been developed specifically for PoS-tagging in Limbu language, we had to create new and original full scale domain specific languages. A corpus is just a huge, well-formatted text documents. This corpus was useful because PoS-tagging (the method that classifies words within a text as nouns, verbs, adjuncts etc.), is data hungry so the ability to learn from examples in context can be very valuable. After that, PoS system has been implemented on various obtained texts to see how accurately and robustly it worked. The test data diversity ensures that the system is evaluated under multiple contextual and language situations.

In order to fully reflect the diversity of Limbu use in different modalities and settings, an extremely detailed ensemble (corpus) has been compiled. To make sure that it reflects multiple linguistic registers, nuances and stiles, diversified text set has been used. Given that the primary objective was to have as holistic and efficient dataset for Limbu language, a lot of diverse text types were incorporated in the corpus set.

As for the data collection, it has been decided to focus on newspapers as its main corpus since it would provide a rich view regarding how Limbu is being used in formal and con- temporary settings. Newspapers are excellent to read because they still use a different level of formality language, and the newspapers reflect contemporary linguistic trends in terms of vocabulary and subject matter. The result of this is that the PoS-tagging system does not find it difficult to adapt a newer formalized usage with ease in Limbu language.

This research thus expanded the corpus to cover instructional material, which provides with instances of how the Limbu language is structured and standardized. These books contain sentences that follow grammatical rules and standards, making them a great resource for training systems to correctly respect formal language. These texts are included to help the PoS-tagging system correctly identify and be able to process the Limbu grammatical structures.

The corpus also includes examples of informal communication, in addition to formal sources such as social media inter- actions and conversational writings. Such writings demonstrate the colloquial, everyday nature of Limbu use in writing as it actually occurs among people who write and speak casually. Since the advanced flexible PoS-tagging system that can handle full range of sentence differences from formal to informal, incorporation of this kind makes the whole model much more general which is then in turn able to accommodate such informality.

The corpus has been enriched, balanced, and made to be of high quality which is required while building an efficient PoS-tagger for the Limbu language. These texts from different sources have been mined so that the sufficiently large dataset can be created, which can be used in training an auto- mated Limbu-literate (to some extent) system for both formal and informal aspects of this particular indigenous language. If we do not have a balanced dataset, our PoS-tagging model may work well with one type of text, but it might fail for another type.

Table 2: Feature Sources

Sources	Corpus size (in sentences)
Newspaper article [Sikkim Herald]	500
Academic Booklet textual dataset	1000
Formal Communication [Sikkim]	1822
Formal Communication [other places]	678

Table 2 highlights the various sources of features used in the re- search work. Tagged corpus represented as =noun|| =adjective|| =verb || =adjective || =noun || =interjection|| =preposition || =adverb || =noun|| =pronoun|| =adverb || =noun||

A. Challenges In Limbu Pos Tagging

Building an automatic POS tagging system for Limbu can be a difficult undertaking as it involves many linguistic as well as resource-related challenges. First of all, Limbu displays agglutinative morphological properties, in which a root word may have multiple suffixes attached to it, resulting in many surface forms for one structural word. This increases lexical variance and causes data sparsity within the corpus.

Limbu is a low-resource language and not many annotated corpora are available for computational experimentation. The same hold true for the performance of machine learning models on languages with relatively small datasets, where context patterns are hard to learn compared to high resource languages.

And as well, grammatical categories including particles, conjunctions and postpositions occur in flexible syntactic contexts of Limbu language which makes it challenging to disambiguate based on a small number of contextual cues. These difficulties have an impact on the performance of statistical and neural tagging models we evaluate here.

4. Implementation

Limbu PoS-tagger has been developed to evaluate the domain of different modelling paradigms for a morphologically rich low-resource language, three representative approaches were chosen: Hidden Markov Models (HMM), Feedforward Neural Networks (FNN) and Bidirectional Long Short-Term Memory networks (BiLSTM). These models include sequence models based on probabilities, and deep neural architectures that can capture the dependencies between words in context. This comparative analysis presents a way to evaluate the treatment of Limbu's linguistic features across these two paradigms. In the first stages of development, used Hidden Markov Model (HMM) a standard statistical technique for PoS-tagging. HMM was chosen due to its efficiency and simplicity in dealing with sequences. Yet this method is highly flawed, mostly because of its feature independence assumption. This made it impossible for the model to summarize and capture many of the pronominalization phenomena in Limbu that are word- context dependent.

Upon discovering these problems, the work turned to a feed forward neural network for use in development. The HMM had limitations in how it was able to model the relationships between words within a phrase, and this neural approach presented a more powerful and flexible option. This neural network was able to model some of the linguistic structures that proved too difficult for an HMM system, derive in a significant increase in both precision and speed for our PoS-tagger. By combining statistic and neural methodologies, the final PoS-tagger for Limbu language provide a more accurate and robust tagging system which captures unique char- acters of form in this under-resourced language.

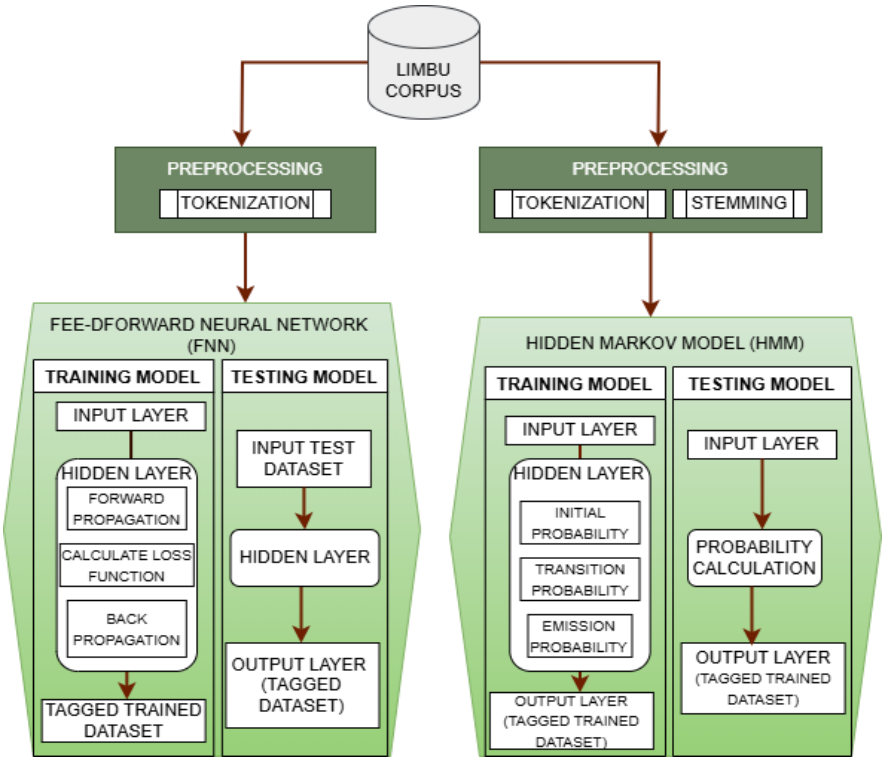

Figure 1. Working Model of PoS-Tagging for Limbu Language.

Systematically, different parts of the Limbu corpus were separated for validating data, testing data and training data. Again, all of this was required so that we can make sure the models are trainable and still test how well they perform, verifying generalizability. Preprocessing the input corpus was initially pre-processed in order to make it analysis. One of these measures was the elimination of stop words, which are a series of common words used in natural language that don‘t conveys meaning like “and”, “the” and so on. Then is the tokenization, which splits up an array of sequences into elements. Furthermore, stemming was used to normalize multiple variations of words by taking them down to their root or base form.

The above pre-processing steps are very important as we have seen this in the preprocessing when it was applied to convert raw data into something that a model can understand which is depicted in figure 1. Afterwards, the pre-processed corpus was fed into two model namely Feedforward Neural Network (FNN) and Hidden Markov Model (HMM). We then used these models on the corpus to give most probable Part-of-Speech (PoS) tags of all the words in the corpus to interpret our data. All the models were able to find success in tagging some of the PoS-tags which allowed all our model variants (which incorporates both statistical and Neural Network approach) to reap their relative strengths and thus add an overall boost for Limbu tagger’s performance.

Feedforward Neural Network (FNN) has been used for Part- of-Speech (PoS) tagging in the Limbu language to benefit from characteristics of this neural architecture. FNN architecture has been chosen as a good model of complex linguistic patterns in Limbu because it could describe intricate relationships between input data and output class. The essential method to help model better learn non-linear correlations in data is using ReLU activation function. ReLU is a very popular choice of activation function for the simple reason that it fixes problems like the vanishing gradients and also helps train extremely deep networks since they are less computationally heavy than other non-linearity functions.

For accurate classification as it is ideal the Cross Entropy has been selected, and knowledge acquisition. This loss function is a common and sensible choice for the discrepancy between ground truths of all classes and their predicted probability distribution, especially in multi-class classification tasks. The strategy employed for improving the FNN to predict PoS-tags of Limbu language consists in making it learn further what a qualified word’s corresponding tag is, which can best minimize Cross Entropy cost. if combined with ReLU and Cross Entropy loss function, the FNN learned well from training data and yielded useful gains in labeling accuracy across multiple languages settings.

The data first goes through a training step in which it is processed with Hidden Markov Model (HMM) trained on an- notated Part-of-Speech tags to be assigned and associated with each word. In training, the model calculated two important probabilities needed for PoS-tagging. Probabilities are the first emission i.e.; a word has specific PoS-tag. Second, they used the transition probabilities to calculate the probability of one PoS-tag preceding another in text. A common approach adopted by the HMM is to use these probabilities in determining what order a sentence should take from a particular result which requires understanding for language's sequential structure.

The Viterbi based PoS-tagger inferred the tags of words in not yet seen texts. We can find the best sequence of states (the actual words themselves) as an observed phenomenon through Viterbi's dynamic programming tool to handle this. The algorithm iterates through each Possible tag sequence and records the likelihood of that being most similar to our input text. With this technique, it performs PoS-tagging to texts in the training set of texts very easily and accurately due to which the model was working great on new language inputs as well.

A. Feed Forward Neural Networks (Fnn)

The training and testing processes of FNN model are performed on a complete set and used corpus of these about 60,000 words. This architecture of the FNN model had three basic components; input layer, hidden layers and output layers. The FNN model had to learn finding all those complex relationships of words and tags by training on this massive corpus.

During an evaluation of the input data, the model changed its weights between neurons in each layer. These weights are the evidence of how strong a link between one word and its expected tag is. This hidden layer had an important function in storing these interactions and adjusting them, which are used to learn the more advanced rules and dependency structures within language. The output layer then produced a number of target tags for the words by converting learnt correlations into an operational format. The efficiency of the model was apprised by matching the ground truth sequence for tags in our testing set to an expected output sequence. Through this, the FNN was able to increase its precision of misclassification in PoS-tag assignment by improving what it understood about language.

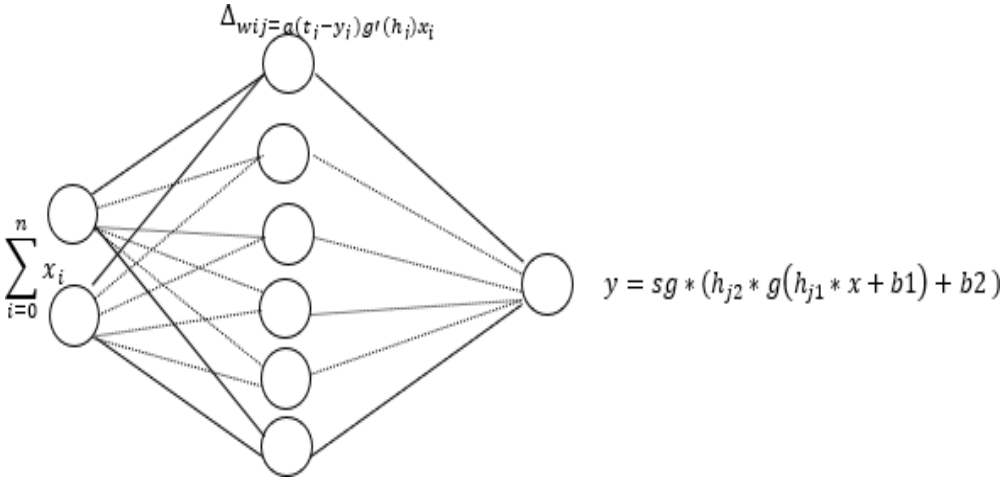

Figure 2. Feed Forward Neural network (FNN) Model.

One implementation of this model, the neural network, has been one of the most significant forces responsible for advances in machine learning as it led to dramatic improvements across wide classes of applications including image recognition and natural language processing. The Feedforward Neural Network (FNN) is the simple normal neural network, or we operate when say single layer, multi-layer architecture etc. Due to the simplicity combined with efficiency, it has become a really useful tool for carrying out many different types of tasks and also processing complex datasets more easily.

FNN model has a different layer such as an input layer, one or more hidden layers and the output layer which act as its structure. So, every layer is mandatory in using incoming data with a useful output. The activation functions in FNN like ReLU which adds non-linearity to the model and allows it to learn complex patterns from data. FNN has trained based on backpropagation and optimization algorithms (for instance, gradient descent), which are applied to improve the weights of a network. So, it better orient FNN in learning and thus predict results. With

more of the FNN algorithm tackle all of the real-time issues described above by understanding these simple functional blocks.

B. Hidden Markov Model (Hmm)

In this paper, we have evaluated the comparison of statistical method i.e. Hidden Markov Model (HMM) and neural network with respect to PoS-tagging in Limbu Language. The HMM is a model in probabilities that can find connections between an observable data sequence (words) and a set of states hidden (tags PoS). The PoS-tags thus represented the hidden states: these internal representations are not directly visible so they are latent but can be inferred based on this observation using a word sequence. To get the sequence of this hidden state and this can be decided by finding out probabilities in HMM that is what order will lead to sequence word.

In this, let xi and yi are the ith word and input though of PoS-tag associated with it. This yields an ordered pair (xi, yi) with each index ranging from 1 to n where n is the total number of words in a sequence. These scores account for the correlations between HMM latent PoS-tag and observation word. Once the model estimates how probable each tag sequence is given that it was emitted by this observation, a Viterbi decoder chooses their best choice as predicted PoS-tags for the provided input text based on which sequence of observations emits them most probabilistically. HMM is a good choice for this kind of problem since it allows the model to learn about the structure and sequencing nature of words being spoken (in other words, HMM helps in retaining some context info), which works well with PoS-tagging. An example of the very probability that HMMs compute is given below:

$$P(x, y) = P(y)P\left(\frac{x}{y}\right) \quad (1)$$

Were, P(y) is a prior probability distribution over tag y. P(x/y) is a conditional probability.

$$P\left(\frac{y}{x}\right) = \frac{P(y)P\left(\frac{x}{y}\right)}{P(x)} \quad (2)$$

Where,

$$P(x) = X\, n \in y\, P(x, n) = X\, n \in y\, P(n)P(x/n) \quad \text{(from (1)) (2)}$$

To derive the relation y = f(x) we have $f(x) = \arg_y^{max}\ P\left(\frac{y}{x}\right)$

From (2), $f(x) = \arg_y^{max}\ \frac{P(y)P\left(\frac{x}{y}\right)}{P(x)}$ $P(x)$ is the independent of y now we have,

$$f(x) = \arg_y^{max} P(y)P\left(\frac{x}{y}\right) \quad (3)$$

Applying first Markov assumption, in a sequence $X_1, X_2, ..., X_n$ $(X_n/X_{n-1}, X_{n-2}, ...X_1) \approx P(X_n/X_{n-1})$

Joint probability using Markov assumption as

$$P(x_1, x_2, ... x_n) = \prod_{i=1}^{n} Px_{i/x_{i-1}} \quad (4)$$

For nth element of the sequence of word

$$f(x_n) = \arg_{yn}^{max} P(yn)P\frac{x_n}{y_n} \quad (5)$$

Assuming, the probability of a word is only dependent on its own PoS-tag, then

$$P(\frac{x_n}{y_n}) \approx \prod_{i=1}^{n} P\left(\frac{x_i}{y_i}\right) \quad (6)$$

which refers to the emission probability matrix. When, probability of PoS-tag is only dependent on the previous PoS-tag, then

$$P(y_n) \approx \prod_{i=2}^{n} P\left(\frac{y_i}{y_{i-1}}\right) \tag{7}$$

which constitutes the transition probability matrix. Using equation (6) and (7) in (5) we have,

$$F(x_n) = \arg_{yn}^{max} \prod_{i=1}^{n} P(\frac{x_i}{y_i}) \prod_{i=2}^{n} P(\frac{y_i}{y_{i-1}}) \tag{8}$$

5. Experimental Results

In this study, the performance of two models has been compared for PoS-tagging on a dataset of Limbu language. Hidden Markov Model (HMM) and Feed Forward Neural Network (FNN) has been used for the PoS-tagging task. The data used in this study is the corpus of a Limbu language advanced and manually annotated using PoS-tagging. As 60/20/20 split ratio, partitioned the dataset into training, validation and testing sets.

For the HMM model Baum-Welch algorithm was used for training and Viterbi algorithm for decoding. Numerous configurations were tested for parameters, including the number of hidden states and the number of iterations in training. Specifically, the FNN model consisted of three-layer neural network, with ReLU activation function and backpropagation-based training. we tried some different network architectures (e.g., different numbers of hidden layer combination) and learning-rate during training.

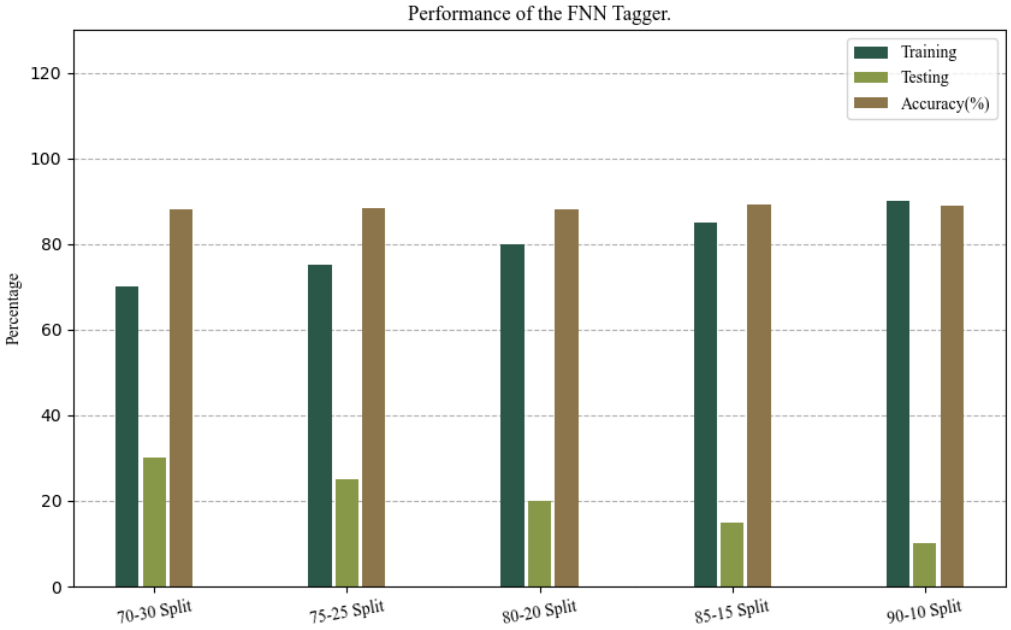

Figure 3. performance of the FNN Tagger.

To test the efficacy of the Feedforward Neural Network (FNN) as regards Part-of-Speech (PoS) tagging, several experiments were conducted using various training and testing dataset size as depicted in figure 3. The dataset was split into different ratios of 70% training to 30% testing to 90% training to 10% testing. We executed each experiment with the corresponding data split, and accuracy of FNN model was measured. The result delightedly closed such that the FNN w.r.t. the 70-30 training-testing split reached an accuracy of 88.16%, which slightly improved with a 75-25 split accuracy of 88.41%. However, there was a drop in accuracy to 88.04% with an 80-20 split. Almost, their performance improved highly to 89.09% and 89.02% accuracy for 85- 15 and 90-10 splits,

while the dataset size was reduced further. Indeed, this indicates the underlying principle that a smaller training dataset along with relevant testing data is powerful enough to derive optimized accuracy in PoS-tagging with the FNN model.

Then we used standard PoS-tagging accuracy parsed from standard metrices (Precision, Recall, F1 Score) to find HMM and FNN model performance scores. Table below highlights the results of our experiments.

Table 3: FNN and HMM Performance on Part of Speech (POS)-Tagging of Limbu Language

Model	Accuracy	Precision	Recall	F1-Score
HMM	0.82	0.83	0.81	0.82
FNN	0.88	0.88	0.88	0.88

Performance of the two powerful PoS-taggers HMM and FNN has been analyzed in the table 3. As shown in the figure 4, both models have high accuracy as well as comparable precision, recall and F1 Scores. While the HMM model was slightly better at identifying predicted words, the FNN model did a better job overall and also achieved an accuracy of 0.88, compared to 0.82 from the HMM model. This is significant because programming languages are particularly difficult to PoS-tag due to their complex syntactic structures and rare words, both of which deserve attention from the FNN model proposed in this approach.

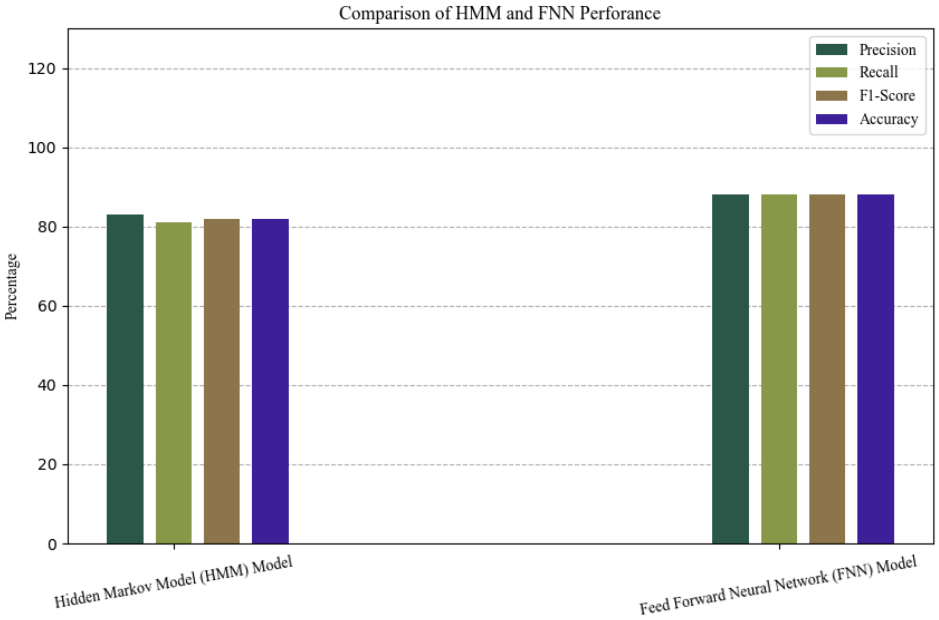

Figure 4. Comparison of FNN and HMM Models based on Different Features.

In summary, our findings imply that the FNN model can potentially be an effective way to perform PoS-tagging for the Limbu language, which often presents challenges due to the complex syntactic structure and use of domain-specific terminology in programming languages, which can greatly benefit from trainable neural approaches.

A. Comparison Among Neural Network Models

In order to evaluate how well the proposed PoS-tagging model performs, we performed a comparison experiment with a Bidirectional Long Short-Term Memory (BiLSTM) model combined with Hidden Markov Model (HMM) and single layer Feed-Forward Neural Network (FNN). BiLSTM is commonly used for sequence-labelling tasks including PoS-tagging because it can model the dependencies not only in past but also in future context of words in a sentence.

Therefore, the BiLSTM model was trained with corpora annotated in a similar way to those used for training of the HMM and FNN models provided with this work for fair comparison. Word-level embeddings were fed to the BiLSTM where the model acquired contextualized knowledge while traversing forward and backward of the sequential data. The output layer was responsible for the prediction of a related PoS-tag that corresponded to each token in the sequence.

Table 4: Comparative Analysis of the Processed Methods for PoS-Tagging of Limbu Language

Model	**Accuracy**	**precision**	**Recall**	**F1-Score**
HMM	0.82	0.83	0.81	0.82
FNN	0.88	0.88	0.88	0.88
BiLSTM	0.89	0.88	0.90	0.89

Table 4 shows the comparative accomplishment of three models in terms of Accuracy, Precision, Recall and F1-score. From the results, we can see that BiLSTM model performs better than HMM and basic FNN in almost all performance measures. This improvement is because BiLSTM has the ability to learn not just from long-distance dependencies but also brings contextual information which plays an important role in morphologically rich and low-resourced languages like Limbu. Although the FNN model shows better results than HMM by learning non-linear features, it does not have explicit sequence modelling," and is therefore insufficient to model contextual tag dependencies.

Despite that the BiLSTM model is more expensive in terms of computation resources and training time, the relative performance improvements demonstrate its potential for PoS-tagging more under-resource languages. However, given the small corpus available for training, the results also indicate that simpler models such as FNN can still be competitive and feasible when there are computational or data restrictions.

Table 5: Representation of Part of Speech (POS)-Tagging of Limbu Text

Pre-processed text as input	Tagged output using FNN
['[illegible]', '[illegible]', '[illegible]', '[illegible]', '[illegible]', '[illegible]', '[illegible]', '[illegible]', '[illegible]', '[illegible]', '[illegible]', '[illegible]', '[illegible]', '[illegible]', '[illegible]', '([illegible])[illegible]', '[illegible]', '[illegible]', '[illegible]', '[illegible]', '([illegible])[illegible]', '[illegible]', '([illegible])[illegible]', '([illegible])[illegible]', '[illegible]']	[illegible] [PRP], [illegible] [VB], [illegible] [VB], [illegible] [JJ], [illegible] [VB], [illegible] [PRP], [illegible] [PRP], [illegible] [VB], [illegible] [VB], [illegible] [PRP], [illegible] [NNS], [illegible] [VB], [illegible] [VB], [illegible] [PRP], [illegible] [NN], ([illegible])[illegible] [VB], [illegible] [PRP], [illegible] [NN], [illegible] [NNP], [illegible] [NNP], ([illegible])[illegible] [JJ], ([illegible])[illegible] [NN], ([illegible])[illegible] [NN], [illegible] [VB]

Table 5 highlights the output on PoS-tagged Limbu corpus and represents how neural methods work on this specific low resource language. The findings indicate that the FNN model, even with limited annotated data and computationally challenging scenario of Limbu, works relatively better. While the accuracy of the BiLSTM model is relatively higher than other models by capturing more sequential and contextual information, this effectiveness is at the expense of additional complexity in computation and learning. The findings validate the use of contextual neural architectures for POS tagging in low resource languages. In contrast to HMM based on transition probabilities and struggling with sparse data, neural approaches like BiLSTM utilize context embeddings, which better capture sequential dependencies. On the other hand, this performance gap reveals that for languages such as Limbu, larger annotated corpora and morphological knowledge are still a crucial aspect to develop. Comparison to other systems FNN is robust and efficient with less resource utilization compared to the others, making it suitable for low resource and also in resource limited environments.

B. Tag-Wise Performance Analysis

To verify the effectiveness of the proposed PoS-tagging framework, tag wise performance was computed evaluated with respect to Precision, Recall and F1-score. This study provides deeper insight into part-of-speech tagging accuracies needed for a morphologically rich low-resource language like Limbu.

Table 6: Tagwise Analysis for Better Comprehension of Limbu PoS-Tagging

PoS Tag	Model	Precision	Recall	F1-score
Noun	HMM	0.82	0.81	0.81
	FNN	0.84	0.86	0.86
	BiLSTM	0.89	0.90	0.90
Verb	HMM	0.86	0.88	0.86
	FNN	0.88	0.88	0.88
	BiLSTM	0.91	0.91	0.91
Adjective	HMM	0.78	0.77	0.78
	FNN	0.80	0.80	0.80
	BiLSTM	0.80	0.81	0.81
Pronouns	HMM	0.72	0.71	0.71
	FNN	0.84	0.85	0.85
	BiLSTM	0.86	0.88	0.88

Major PoS categories (Noun, Verb, Adjective and Pronouns) are analyzed in Table VI. Results indicate that Nouns and Verbs are easier to tag as their F1-scores are higher. This may be related to both a richer distribution of the training data and more salient morphological patterning. In comparison, for Functional Categories the performance is significantly lower, which traces back to their functional status and frequent ambiguity with other grammatical types. Such tags often rely more on context than content features and are therefore more difficult to be accurately classified.

Also, less frequent tags show worse recall: data sparseness impairs the model's ability to generalize on rare grammatical forms. Nevertheless, the ablation studies show that our framework has successfully captured the essential syntactic skeleton of Limbu sentences and have more precisely pinpointed those tags for which larger annotated data or better linguistic features are needed for future work.

C. Hyperparameter Analysis

In order to explore the effect of model capacity on tagging performance, we performed additional experiments with varying numbers of hidden units in BiLSTM architecture.

Table 7: Accuracy Analysis with Varying units of BiLSTM Model

Model	**Hidden Layers**	**Accuracy (%)**
BiLSTM	62	88.8
BiLSTM	124	89.56
BiLSTM	248	89

As Table 7 displays, having more hidden units, 124 compared to the lower bound of 62 increases tagging accuracy from 88.8% to 89.56%. This enhancement is due to the model's improved capacity to grasp contextual dependencies and sequential patterns within the data.

But doubling the hidden states to 248 does not improve further and reduces accuracy to 89%. This behavior can be attributed to overfitting of the training corpus, which is small in comparison to a full knowledge corpus. While larger models generally require more training data to generalize well, and large architectures can lead to underfitting in low-resource settings such as Limbu.

From these observations, 124 hidden units provide the best result between model complexity and generalization ability over the dataset used. Hence, the model with 124 hidden units of BiLSTM was chosen as the optimal configuration for further experiments.

D. Analysis Of An Error

Only a qualitative analysis of tagging errors shows that the large majority of misclassifications occur in functional categories particles, conjunctions and postpositions. These categories often do occur in syntactically flexible contexts, making it difficult to disambiguate between them without additional contextual clues. For instance, Limbu has particles which serve as discourse markers or grammatical modifiers based on their positioning within a particular structure.

Unlike HMM and FNN, other neural models such as BiLSTM model bidirectional contextual dependencies which accounts to better performance for these cases. Conversely, the aforementioned errors still occur when rare morphological variants are present in the dataset effectively showing the limitations brought on by a comparatively small training corpus.

6. Conclusion

In this paper, an analysis has been conducted on PoS-tagging for the Limbu language, applied to traditional and neural methodology Hidden Markov model (HMM) and Feedforward Neural Network (FNN) respectively. The HMM well modeled sequential characteristics of Limbu text based on probabilistic approach and the FNN exhibited excellent learning ability when two stages were applied to select important features. For additional confirmation to the proposed architecture, we compared with BiLSTM model and found that HMM and FNN are still efficient and effective in low-resource scenario, but its overall performance is significantly improved by leveraging bidirectional context from original Limbu text sentences.

This work contributes to the understanding of Limbu language processing by characterizing the strengths and weaknesses of HMM, FNN, and BiLSTM models directly while providing strong baseline for the future. Future work can concentrate on extending the HMM by higher-order linguistic knowledge or investigating better neural models for PoS-tagging. The combination of this kind of classical probabilistic modeling and more recent deep learning techniques provides enormous potential for the advancement of natural language processing in under-resourced languages thus contributing to better Limbu understanding, cross-cultural communication and technological progress.

References

1. P. Srivastava, K. Chauhan, D. Aggarwal, A. Shukla, J. Dhar, and V. P. Jain, “Deep learning based unsupervised PoS tagging for Sanskrit,” in Proc. ACAI 2018, Sanya, China, Dec. 2018. Association for Computing Machinery, doi: 10.1145/3302425.3302487
2. K. Toutanova, D. Klein, C. D. Manning, and Y. Singer, “Feature-rich part-of-speech tagging with a cyclic dependency network,” in Proc. HLT-NAACL 2003, Edmonton, Canada, 2003, pp. 173–180.
3. C. Ding, H. T. V. Jatav, R. Teja, S. Bharadwaj, and V. Srinivasan, “Improving part-of-speech tagging for NLP pipelines,” arXiv preprint arXiv:1708.00241 [cs.CL], 2017.
4. W. P. Pa Aye, K. T. Nwet, K. M. Soe, M. Utiyama, and E. Sumita, “Towards Burmese (Myanmar) morphological analysis: Syllable based tokenization and part-of-speech tagging,” ACM Transactions on Asian and Low-Resource Language Information Processing, vol. 19, no. 1, Art. no. 5, May 2019.
5. F. M. Hasan, N. UzZaman, and M. Khan, “Comparison of different PoS tagging techniques (N-Gram, HMM and Brill’s tagger) for Bangla,” in Advances and Innovations in Systems, Computing Sciences and Software Engineering, 2007.
6. S. Parikh, “Part-of-speech tagging using neural network,” in Proc. ICON 2009, 2009.
7. S. Chotirat and P. Meesad, “Part-of-speech tagging enhancement to natural language processing for Thai wh-question classification with deep learning,” Heliyon, vol. 7, no. 10, Art. no. e08216, 2021.
8. 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, 2015.

9. M. Shrivastava and P. Bhattacharyya, “Hindi PoS tagger using naive stemming: Harnessing morphological information without extensive linguistic knowledge,” in Proc. ICON 2008, Macmillan Publishers, 2008.
10. S. Tyagi and G. S. Mishra, “Statistical analysis of part-of-speech (PoS) tagging algorithms for English corpus,” International Journal of Advance Research (IJARIIT): Ideas and Innovations in Technology, vol. 2, no. 3, 2016.
11. H. Kaing, C. Ding, M. Utiyama, E. Sumita, S. Sam, S. Seng, K. Sudoh, and S. Nakamura, “Towards tokenization and part-of-speech tagging for Khmer: Data and discussion,” ACM Transactions on Asian and Low-Resource Language Information Processing, vol. 20, no. 6, Art. no. 104, pp. 1–16, Sep. 2021.
12. Y. Yajnik, “Part of speech tagging using statistical approach for Nepali text,” World Academy of Science, Engineering and Technology International Journal of Cognitive and Language Sciences, vol. 11, no. 1, 2017.
13. D. Ding, M. Utiyama, and E. Sumita, “NOVA: A feasible and flexible annotation system for joint tokenization and part-of-speech tagging,” ACM Transactions on Asian and Low-Resource Language Information Processing, vol. 18, no. 2, Art. no. 17, Dec. 2018.
14. A. Jamatia, B. Gambäck, and A. Das, “Part-of-speech tagging for code-mixed English–Hindi Twitter data,” in Proc. ICON 2015, 2015, pp. 239–248.
15. R. Mandal, V. Singh, and D. Singh, “A survey on NLP tasks, resources and techniques for low-resource Telugu–English code-mixed text,” Artificial Intelligence Review, vol. 55, pp. 6481–651, 2022, doi: 10.1007/s10462-022-10179-1.
16. H. Tang, Y. Jiang, Y. Zhang, N. Niu, and H. Liu, “PoS tagging on code identifiers: How far are we?” in Proc. IConSCEPT 2024, IEEE, 2024, doi: 10.1109/IConSCEPT61884.2024.10627782.
17. D. Baishya and R. Baruah, “Part-of-speech tagging for low-resource languages: Activation function for deep learning network to work with minimal training data,” ACM Transactions on Asian and Low-Resource Language Information Processing, vol. 23, no. 5, Art. no. 70, pp. 1–31, May 2024, doi: 10.1145/3655023.
18. S. K. Nambiar, D. Peter S., and S. M. Idicula, “Abstractive summarization of text document in Malayalam language: Enhancing attention model using PoS tagging feature,” ACM Transactions on Asian and Low-Resource Language Information Processing, vol. 22, no. 2, Art. no. 59, pp. 1–14, Mar. 2023, doi: 10.1145/3561819.
19. S. Warjri, P. Pakray, S. A. Lyngdoh, and A. K. Maji, “Part-of-speech (PoS) tagging using deep learning-based approaches on the designed Khasi PoS corpus,” ACM Transactions on Asian and Low-Resource Language Information Processing, vol. 21, no. 3, Art. no. 63, pp. 1–24, Dec. 2021, doi: 10.1145/3488381.
20. N. Bölücü and B. Can, “Unsupervised joint PoS tagging and stemming for agglutinative languages,” ACM Transactions on Asian and Low-Resource Language Information Processing, vol. 18, no. 3, Art. no. 25, pp. 1–21, Jan. 2019, doi: 10.1145/3292398.
21. H. Wang, J. Yang, and Y. Zhang, “From Genesis to Creole language: Transfer learning for Singlish universal dependencies parsing and PoS tagging,” ACM Transactions on Asian and Low-Resource Language Information Processing, vol. 19, no. 1, Art. no. 1, pp. 1–29, May 2019, doi: 10.1145/3321128.
22. Y. Li, X. Li, Y. Wang, H. Lv, F. Li, and L. Duo, “Character-based joint word segmentation and part-of-speech tagging for Tibetan based on deep learning,” ACM Transactions on Asian and Low-Resource Language Information Processing, vol. 21, no. 5, Art. no. 95, pp. 1–15, Nov. 2022, doi: 10.1145/3511600.
23. T. Dalai, T. K. Mishra, and P. K. Sa, “Deep learning-based PoS tagger and chunker for Odia language using pre-trained transformers,” ACM Transactions on Asian and Low-Resource Language Information Processing, vol. 23, no. 2, Art. no. 19, pp. 1–23, Feb. 2024, doi: 10.1145/3637877.
24. D. Pathak, S. Nandi, and P. Sarmah, “Part-of-speech tagger for Assamese using ensembling approach,” ACM Transactions on Asian and Low-Resource Language Information Processing, vol. 22, no. 10, Art. no. 235, pp. 1–22, Oct. 2023, doi: 10.1145/3617653.
25. H. Visuwalingam, R. Sakuntharaj, and R. G. Ragel, “Part of speech tagging for Tamil language using deep learning,” in Proc. ICIIS 2021, IEEE, 2021, doi: 10.1109/ICIIS53135.2021.9660738.
26. H. Visuwalingam, R. Sakuntharaj, J. Alawatugoda, and R. Ragel, “Deep learning model for Tamil part-of- speech tagging,” The Computer Journal, vol. 67, no. 8, pp. 2633–2642, Aug. 2024, doi: 10.1093/comjnl/bxae033.