Journal of King Saud University – Computer and Information Sciences 34 (2022) 8739–8749

Contents lists available at ScienceDirect

Journal of King Saud University – Computer and Information Sciences

journal homepage: www.sciencedirect.com

Hierarchical self attention based sequential labelling model for Bhojpuri, Maithili and Magahi languages

Rajesh Kumar Mundotiya [a,*], Swasti Mishra [b], Anil Kumar Singh [a]

[a] Department of Computer Science and Engineering, Indian Institute of Technology (BHU), Varanasi, India
[b] Department of Humanistic Studies, Indian Institute of Technology (BHU), Varanasi, India

ARTICLE INFO

Article history:
Received 26 July 2021
Revised 5 September 2021
Accepted 25 September 2021
Available online 7 October 2021

Keywords:
Datasets
Machine learning
Neural network
POS tagging
Chunking
Transfer learning

ABSTRACT

Sequential labelling plays a vital role in solving numerous Natural Language Processing (NLP) applications such as Machine Translation and Information Extraction etc. One of these is Part-of-Speech (POS) tagging, which assigns a sequence of grammatical categories to the given sentence, and Chunking which groups them into 'chunks' or what can be called minimal phrases. Bhojpuri, Maithili and Magahi are low resource languages and widely spoken in central north-eastern India, belonging to the Indo-Aryan language family. The creation of an annotated corpus for POS tagging and Chunking, and then building an initial automatic tool for these problems is the first attempt towards building language technology tools for these languages. The annotated corpus used to develop POS Taggers and Chunkers, based on various machine learning algorithms (TnT, CRF, MEMM and Structured SVM) and state-of-the-art LSTM-CNN-CRF model, and then these compared with the obtained results on two new proposed deep learning-based models, Self-Attention Hierarchical Bi-LSTM CRF (SAHBiLC) and a fine-tuned version of it, Fine-SAHBiLC. The SAHBiLC and Fine-SAHBiLC models outperform on Bhojpuri (Accuracy for POS and Chunking is 0.86% and 0.94%, respectively) and Maithili (Accuracy for POS and Chunking is 0.86% and 0.95%, respectively) and Magahi (Accuracy for POS is 0.86%).

1. Introduction

With the relatively sudden and so far enduring success of deep neural network based approaches, Artificial Intelligence (AI) technologies have come to play a vital role in the individual and collective lives of people across the world. Along with the new algorithms, the abundance of data for machine learning and the development of suitable computing resources, this almost radical change has affected AI problems. Natural Language Processing (NLP) is one of the fields taking advantages of new AI approaches. Recent fast development of methodologies and models have substantially affected different tasks, like word and sentences labelling, and have allowed new technologies to exceed the previous state-of-the-art results.

Part-of-speech (POS) tagging assigns a sequence of grammatical categories (POS tags) to the given word sequence (sentence), while Chunking links POS tagged words into groups of words or 'chunks', which can be roughly defined as minimal phrases or minimal constituents. Both of these tasks are often performed in the preliminary stages of any complex language processing task. Thus, for any new language, i.e., a language that does not yet have language tools developed for it, it is crucial to first build applications for performing these two tasks for that language. The two tasks are usually part of a Natural Language Processing (NLP) pipeline and play a significant role in various more complex tasks such as Named Entity Recognition, Question Answering, Information Extraction, Machine Translation and so forth. There are more than 7000 natural languages[1] that are still widespread use in the world. It is important to remember that most of these languages are low resource languages or resource scarce languages (Christianson et al., 2018). This fact becomes even more important and relevant if we consider that many of these low resource languages are among the most

* Corresponding author.
E-mail addresses: rajeshkm.rs.cse16@iitbhu.ac.in (R.K. Mundotiya), swasti.hss@iitbhu.ac.in (S. Mishra), aksingh.cse@iitbhu.ac.in (A.K. Singh).
Peer review under responsibility of King Saud University.

Production and hosting by Elsevier

[1] https://www.ethnologue.com/guides/how-many-languages.

https://doi.org/10.1016/j.jksuci.2021.09.022
1319-1578/

spoken languages of the world. This is particularly true of the languages of South Asia. Majority of Indian (or South Asian) languages do not yet have as large or advanced language resources or language processing applications as we see for most European languages. Deep neural networks are a general paradigm to solve language processing problems, but their applicability to low resource languages is hampered by the fact that they need large quantities of appropriate data (usually with some kind of human annotation). Still, deep learning-based approaches have been used to develop language technology tools for low resource language (Singh et al., 2008; Saha et al., 2008; PVS and Karthik, 2007). This is made possible by various advances in such approaches that address the problem of resource sparsity.

As an example of advances in deep neural networks which address the sparsity issue, Transfer Learning techniques (Yang et al., 2017; Kim et al., 2017) work well in a scenario where an ample amount of annotated data of source language is available, but not of the target language data, which is the low resource language. A large amount of annotated data of the source language which is resource rich from related tasks can help in enhancing the performance of deficient annotated data of the target language. The advantage of Transfer Learning is, thus, that it does not require additional resources for the target language to mitigate data sparsity as explored in the following section on related work. The deep learning model for the same task, say POS tagging, is trained on both the source and target language, after that the model from the source language data is partially transferred to the target language by way of sharing the hidden representation (weights) between the two languages.

In our case, Bhojpuri, Maithili and Magahi are the target languages, which are closely related to Hindi as the source language. Although Hindi is still not as resource rich as some of the much less spoken European languages, it is still relatively richer to the extent that we can get some useful results. Like most Indian languages, all these four languages have SOV word order (Subbārāo, 2012), and they all use the Devanagari script for their writing systems. Linguistically, they belong to the same sub-family of Indo-Aryan (IA) language family. In fact, till recently, the three concerned languages were considered as dialects or variants of Hindi. This makes for an ideal situation for model transfer for POS tagging and Chunking.

In this paper, we first train conventional machine learning algorithms for POS tagging and chunking by using handcrafted features. Later, we compare the results with deep learning based current baseline technique using monolingual embeddings. We find that Transfer Learning-based indeed outperforms conventional machine learning for these languages on POS tagging and Chunking. These results show the potential of leveraging the Hindi model's parameters could help many other languages, which are also similar to Hindi. We further investigate if it is possible to get more improvement on the sequence labelling problem by fine-tuning the disambiguation layer.

1.1. Contribution

Our contributions are as follows in this article:

- Build a more accurate deep learning based POS taggers compared to previously developed taggers for Bhojpuri, Maithili and Magahi languages with greater data size.
- First work towards developing Chunker for Bhojpuri and Maithili languages.
- Improve POS tagger and Chunker performance for these languages, using Hindi as a high-resource language through cross-lingual (learnt) feature transfer and discuss the case of negative transfer.

2. Related work

In recent works, deep learning-based techniques have populated due to independence on traditional features such as unknown (rare frequency) word, lexicon, affixes, available words and tags context for sequence labelling tasks such as POS tagging, Chunking and Named Entity Recognition.

Convolutional Neural Network (CNN) along with character level's word representation for POS tagging, is explored by dos Santos and Zadrozny (2014). The CNN is unable to capture the temporal relations; consequently, it neglects the dependencies among the words of the sentences. Recurrent Neural Network (RNN) with its varients Gated Recurrent Unit (GRU), Long Short Term Memory (LSTM), Bidirectional-RNN (Bi-RNN), Bidirectional-LSTM (Bi-LSTM), Bidirectional-GRU (Bi-GRU) and stacking proposed to capture features of longer dependencies for each word.

At the very beginning, Huang et al. (2015) used the LSTM unit as a word information encoder with CRF as a label sequence decoder. Ma et al. (2016) extend Lample et al. (2016) work, which was proposed for Named Entity Recognition, by using CNN to capture intra-word information used with word embedding. The word sequence and label sequence information were captured and decoded by Bi-LSTM and CRF, respectively. Dozat et al. (2017) used unidirectional LSTM for character-level information followed by a linear attention mechanism. Zhang et al. (2018) proposed a multi-channel model based on the Bi-LSTM for obtaining the word and label dependencies and their interaction simultaneously by using the label of the previous word as a context for the current word in the softmax decoder.

Kann et al. accomplished this work with the help of three auxiliary tasks a) lemmatization, b) character-based random string autoencoding, and c) character-based word autoencoding. POS tagging for low-resource language takes advantage of the raw text autoencoding, using the lemma information. In low-resource scenarios, the neural POS tagger closes the gap to the state-of-art POS tagger (single task), even exceeding it on languages with templatic morphology (e.g., Arabic, Hebrew, and Turkish) to a significant degree (Kann et al., 2018).

Plank et al. used a multi-task Bi-LSTM model with auxiliary loss and they evaluated tokens and sub-levels for neural network-based POS tagging. The auxiliary loss can be used to improve the accuracy of rare words (Plank et al., 2016). Mishra et al. used feature transfer from a rich-resource language to resource-poor languages, without any knowledge of the target language and human annotation (Mishra et al., 2017).

The first reported attempt towards the development of deep learning-based Maithili POS tagger was trained on a continuous bag of word model (CBOW), word embedding trained with the help of available web resources and Wikipedia dump as corpus using the embeddings (Priyadarshi and Saha, 2020). As such, no substantial state-of-the-art work on sequential labelling problems for Purvanchal languages has attempted using deep learning techniques to the best of our knowledge.

3. SAHBiLC: self attention based hierarchical Bi-LSTM CRF

For the deep learning approaches, words are described as real-valued vectors in a low dimensional semantic space usually known as a continuous vector space. The conventional way for such representations is to compute the term-document occurrence matrix on large corpora and then reduce the dimensionality of a matrix by singular value decomposition (Bullinaria and Levy, 2012; Deerwester et al., 1990; Turian et al., 2010; Collobert and Weston, 2008). Over the recent past years, that conventional two-phase approach has been replaced by a single supervised or

unsupervised method, usually based on neural networks (Levy and Goldberg, 2014).

The information about word morphology is not explicitly considered while building word representations, because learning takes place to only capture syntactic and semantic information through corpus-based distribution. However, intra-word information is intensely useful, especially when the language is morphologically rich and particularly for sequential processing such as part-of-speech tagging (Santos and Zadrozny, 2014). In low resource languages, appearance of unknown words or out-of-vocabulary (OOV) words or tokens (such as dates and times) is an obvious problem that can be handled by character level embedding, where each word is represented as a sequence of characters.

We call our model Self Attention based Hierarchical Bi-LSTM CRF (SAHBiLC), which follows the hierarchical RNN-CRF as a base architecture (Yang et al., 2017). The SAHBiLC has three components which capture the sub-word information, sensitive contextual information and label dependencies, as shown in Fig. 1. The character level Bidirectional LSTM aids to elicit morpho-semantic information and affix information in a hidden representation without explicitly being fed to the network. The word level neural network also implemented through bidirectional LSTM, can reveal sensitive syntactic information and extract dependency relations based on the self-attention layer from the input sentence, which extends the base architecture. A linear chain CRF helps to resolve the dependency among labels and produces inference.

Given the input sequence of words $x_1 \dots x_i$, and character sequence corresponding to each word $c_1 \dots c_n$, the model first takes the word x_i as input and extracts continuous representation through a distribution embedding. The input for character level $\overrightarrow{LSTM}$ is the character sequence corresponding to the word, each character encoded into continuous representation through neural network embedding layer. The output from the end node of character level LSTM concatenates with word representation and admits it as the final word representation, which includes the sub-word information. The sequence of POS tags $t_1 \dots t_i$ corresponding to the sentence is concatenated with the word embedding as an additional input for Chunking.

$$w_i = x_i \oplus \overrightarrow{LSTM}(c_1 \dots c_n)_i \tag{1}$$

$$w_i = x_i \oplus \overrightarrow{LSTM}(c_1 \dots c_n)_i \oplus t_i \tag{2}$$

The word level Bidirectional LSTM layer works as a disambiguation layer (Murthy et al., 2018) for sequence labelling task. Final word representation fed to the forward and backward LSTM layer and concatenates resultant representation for each word, and it holds the sentence-level syntactic information.

$$h_i = \overrightarrow{LSTM}\left(w_i, \overrightarrow{h_{i-1}}\right) \oplus \overleftarrow{LSTM}\left(w_i, \overleftarrow{h_{i-1}}\right) \tag{3}$$

Self-attention (Vaswani et al., 2017) produces a context vector for each word of the sentence which is independent of word positions. It provides flexibility concerning word order and helps to generalise better to hold contextual meaning. Each word's previous output (bidirectional LSTM) is multiplied by the Key, Query, and Value weight vectors. Xavier initializes these weight vectors. Attention scores have been generated after performing product operations between each Query and all Keys obtained, over which softmax is performed. The generated softmax attention score for each word is multiplied by its corresponding Value, and these weighted values are summed to obtain the context weight vector.

The output from self-attention, i.e., context weight vector merges with disambiguation layer of the hidden representation. The CRF layer is employed after the self-attention layer and it is fed concatenated hidden representation:

$$H = [h_i \dots h_i] \tag{4}$$

$$h_i = Self\ Attention(H) \tag{5}$$

$$y\prime = argmax_y \left[\prod_{i=1}^{T} exp\left(\sum_{k=1}^{K} \lambda_k f_k(y_{i-1}, y_i, h_i)\right)\right] \tag{6}$$

4. Training settings

4.1. Dataset

To conduct the tagging experiments with Purvanchal languages, we used them as a low resource languages (target languages), while Khari Boli (Standard Hindi) was used as the high resource language (source language). Purvanchal languages' dataset from Mundotiya et al. (2021) and Hindi dataset from Tandon et al. (2016) are used here for the task of POS tagging and Chunking. The annotated data was tagged with the BIS tagset for Indic languages. Chunk annotated dataset is not available for the Magahi, and Hindi utilised to correlate the obtained results. The basic statistics of the corpus of each language are mentioned in Table 1. For example, the maximum sentence length of Bhojpuri, Maithili and Magahi is 118, 272, 109 words respectively. The most common POS tags for the three languages were found to be NN (noun), VM (main verb), PSP (pronoun) and SYM (symbol, or roughly punctuation), and they covers 50% the annotated data, whereas CL (classifier), ECH (echo word), UNK (unknown word) and UT (quotative) are frequent, but their use is more subtle. From this dataset, we remove those sentences whose length is less than two as part of preprocessing for both the learning techniques.

The proposed deep learning based model, SAHBiLC has compared with the machine learning techniques which have become popular over time, such as Trigrams 'n' Tags (which is an extended variation of the HMM model, using interpolated smoothing), Maximum Entropy Markov Model, Conditional Random Fields and Structured Support Vector Machine, have been using for sequence tagging on Bhojpuri, Maithili and Magahi.

4.2. Machine learning strategy

The preprocessed annotated dataset is divided into a 1:3 ratio for training and validation set. The experiment was conducted using k-fold cross-validation, where 10 is the value for k, with a random sample generator through random seed to provide robustness and prevent overfitting. We conducted experiments with different training-validation ratios, as they require external features, and this helps us get more reliable predictions and use the learned model for evaluation. CRF, SVM and MaxEnt are feature-based techniques. We prepare a default feature set for these techniques, as shown in Table 2.

The CRF technique is trained on gradient-based L-BFGS algorithm with L1 and L2 regularisation till 100 iterations and we retain the remaining parameter as the default. Similarly, we have taken the same number of iterations with MaxEnt, and the threshold value for a rare word is 10. We use TnT and SVM with default settings is available on NLTK and SVMTool, respectively.

4.3. Deep learning strategy

The SAHBiLC model takes word and character as input for POS tagging, while POS information used as input only for Chunking. For the SAHBiLC model, we make inputs in equal length and eliminate the OOV tokens at word and character level by adding special tokens that are $\langle PAD \rangle, \langle UNK \rangle$. The similar length characters and

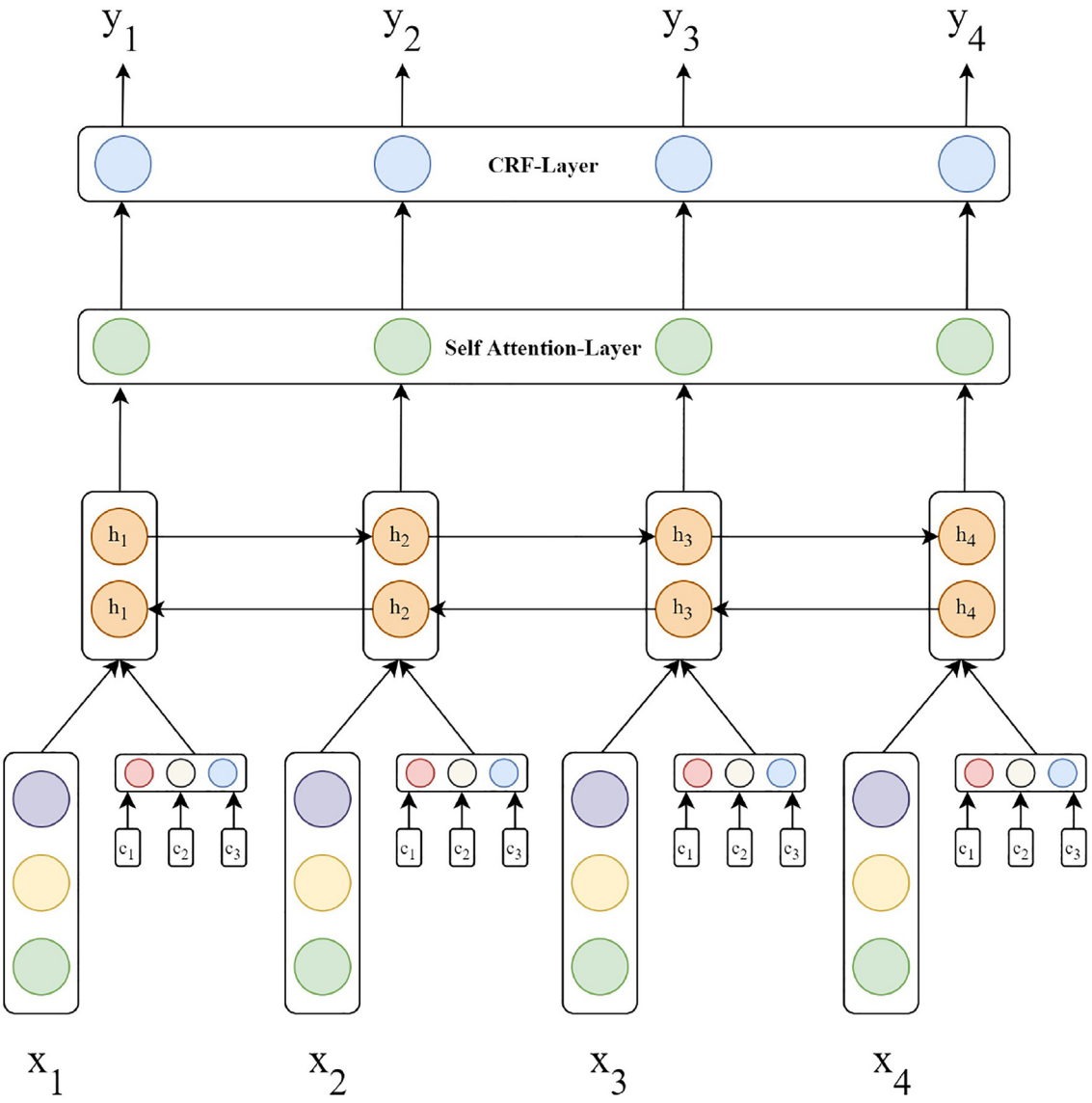

Fig. 1. SAHBiLC model architecture for POS Tagging.

Table 1
Annotation statistics of POS taging and Chuking datasets.

Language	POS annotation			Chunk annotation			
	# Sentence	# Token	# Types	# Sentence	# Token	# Types	# Chunks
Bhojpuri	16067	245482	26202	9695	60588	18090	40239
Maithili	12310	208640	21410	1954	10476	5764	10436
Magahi	14669	171509	14077	–	–	–	–
Hindi	20783	434856	22171	20783	434856	22171	233864

words have transformed into a 10-dimensional vector and 80 dimensions for word vector by distributional representation, respectively. For Chunking also, the POS labels have been represented into a 200-dimensional vectors. A 10-dimensional character vector helps to generate word embedding after applying LSTM with 120 hidden units. These word vector and word embedding perform disambiguation with 200 hidden units of Bi-LSTM, and 0.2 dropouts to reduce the ambiguity. The contextual weightage assigned by multiplicative self-attention, regularized by L1 bias and L2 kernel. The whole training time takes 32 samples in each step and iterates to 10 epochs. For Fine-SAHBiLC model, the vocabularies of character and word from low and high resource language merge together to create a shared embedding space for transferring the learned parameter as a base parameter, instead of random initialization. We employ Adam optimizer with a learning rate of 0.001 for performing the training strategies with the parameter as mentioned above and hyper-parameter for both the models. The parameter and hyper-parameter are summarized in Table 3.

4.3.1. Feature transfer

Monolingual extended hierarchical RNN-CRF model labelling quality can be improved while using an ample amount of training

Table 2
Feature set used for POS Tagging and Chunking in machine learning techniques and contextual boundary value is up to 3, represented by *j*.

Category	Feature	Feature Values
POS Tagging	Word Position	i
	+ Prefix's	p_{i-1}, p_i, p_{i+1}
	+ Suffix's	s_{i-1}, s_i, s_{i+1}
	+ Prefix's length	$p_{i-1_j}, p_{i_j}, P_{i+1_j}$
	+ Suffix's length	$s_{i-1_j}, s_{i_j}, s_{i+1_j}$
	+ Contextual Word	w_{i-1}, w_{i+1}
	+ Contextual Word length	$w_{i-1_j}, w_{i_j}, w_{i+1_j}$
	+ Digits	d_{i-1}, d_i, d_{i+1}
	+ Punctuation	.?!|
Chunking	POS Tagging	
	+ Contextual POS tag	t_{i-1}, t_{i+1}
	+ Contextual POS tag length	$t_{i-1_j}, t_{i_j}, t_{i+1_j}$
	+ Hyphenated	–
	+ Symbol	, -

Table 3
Parameters and Hyper-parameters employed during training the SAHBiLC model.

Parameters and Hyper-parameters	Values
Batch size	32
Epoch	10
Char embedding	10
Word embedding	80
POS embedding	200
Char LSTM unit	120
Word Bi-LSTM unit	200
Dropout	0.2
Attention	Multiplicative
Optimizer	Adam
Learning rate	0.001

data of another language which could be a high resource language. Here, we use Hindi as the high resource language for our sequential labelling tasks for Bhojpuri, Maithili and Magahi. The orthographic system of Hindi, Bhojpuri, Maithili and Magahi languages are very similar, and most of the characters are common, as all three languages use the popular Devanagari script.

We share the character at character level Bidirectional LSTM between Hindi to the corresponding language, and it should disambiguate the Hindi word features from corresponding language word features induced from this layer. The disambiguation layer fine-tuned for each low resource language from Hindi. This fine-tuned model is named as Fine-SAHBiLC model. After sharing the sub-word information, the learnt weights of Hindi are used to initialisation for low resource language. This helps to improve the performance of the sequence labelling task.

5. Result

In this section, we have explained the experimental result at token level obtained on the test set. The stated result of machine learning techniques depends on feature sets for POS tagging and Chunking, as mentioned in Table 2. The performance of each system is evaluated in terms of Precision, Recall, F-score and Accuracy as weighted average.

Machine learning techniques applied to POS tagging and Chunking provides satisfactory result for all three languages; CRF outperforms for all four languages. The comparative result of POS tagging and Chunking at different scales after applying diverse machine learning techniques mentioned in Table 4 and Table 5, respectively. For Chunking, CRF model performs the best for Bhojpuri, Maithili and Hindi with 0.95, 0.94 and 0.99 respectively.

A comparative performance was obtained from SVMTool and CRF for POS tagging on less frequent (INJ, CL, ECH, UNK, UT) and most frequent (NN, VM, PSP, SYM) tags of Bhojpuri, Maithili and Magahi. The F-score for each tag on Bhojpuri, Maithili and Magahi, each technique is shown in the Fig. 2.

Only the TnT technique was able to predict the VGINF chunk tag in Bhojpuri, whereas the remaining less frequent (RBP) and most frequent (NP, VGF, CCP) chunk tags were more correctly predicted by CRF as compared to other techniques. Similarly, Maithili chunk tag (for less frequent and most frequent) was accurately predicted by CRF. The F-score for each chunk tag on Bhojpuri and Maithili, each technique is shown in the Fig. 3.

The proposed **Deep learning** based model for the POS tagging and Chunking has been compared with the state-of-the-art model, LSTM-CNN-CRF (Ma et al., 2016) and Hindi dataset. The results for deep Learning based techniques for POS tagging are quite interesting because SAHBiLC model performs better for Bhojpuri, while Fine-SAHBiLC performs better for Maithili and Magahi, as shown in Table 6. After observing Table 7 for Chunking, SAHBiLC model performs better for Bhojpuri and Fine-SAHBiLC for Maithili. After comparing the entire stated result at F-score for POS tagging, we get Bhojpuri, Maithili, Magahi with 0.87 on SAHBiLC model, 0.86 on Fine-SAHBiLC, 0.86 on Fine-SAHBiLC respectively. Our Fine-SAHBiLC model performs well for these languages where data size is low such as Magahi and Maithili, compared to Bhojpuri. Deep learning-based techniques require a large amount of data to get efficient features. Not Hindi, but Bhojpuri, Maithili and Magahi

Table 4
Results of traditional machine learning techniques (%) in terms of **A**ccuracy, **P**recision, **R**ecall and **F**-score for POS Tagging.

Language	Technique	A	P	R	F
Bhojpuri	TnT	0.83	0.83	0.83	0.83
	CRF	0.86	0.86	0.86	0.86
	MaxEnt	0.83	0.83	0.83	0.83
	SVMTool	0.85	0.87	0.85	0.86
Maithili	TnT	0.80	0.85	0.81	0.83
	CRF	0.85	0.86	0.85	0.85
	MaxEnt	0.84	0.84	0.84	0.84
	SVMTool	0.84	0.86	0.84	0.85
Magahi	TnT	0.81	0.84	0.81	0.82
	CRF	0.83	0.83	0.83	0.83
	MaxEnt	0.81	0.82	0.82	0.82
	SVMTool	0.80	0.83	0.81	0.81
Hindi	TnT	0.93	0.95	0.93	0.94
	CRF	0.94	0.94	0.94	0.94
	MaxEnt	0.93	0.94	0.94	0.94
	SVMTool	0.92	0.93	0.92	0.92

Table 5
Results for traditional machine learning techniques in terms of **A**ccuracy, **P**recision, **R**ecall and **F**-score for Chunking.

Language	Technique	A	P	R	F
Bhojpuri	TnT	0.67	0.76	0.67	0.71
	CRF	0.95	0.94	0.95	0.95
	MaxEnt	0.92	0.92	0.93	0.92
	SVMTool	0.94	0.93	0.93	0.94
Maithili	TnT	0.69	0.80	0.70	0.74
	CRF	0.94	0.94	0.95	0.94
	MaxEnt	0.91	0.91	0.92	0.91
	SVMTool	0.92	0.93	0.93	0.93
Hindi	TnT	0.95	0.94	0.95	0.94
	CRF	0.99	0.99	0.99	0.99
	MaxEnt	0.97	0.97	0.98	0.97
	SVMTool	0.98	0.98	0.99	0.98

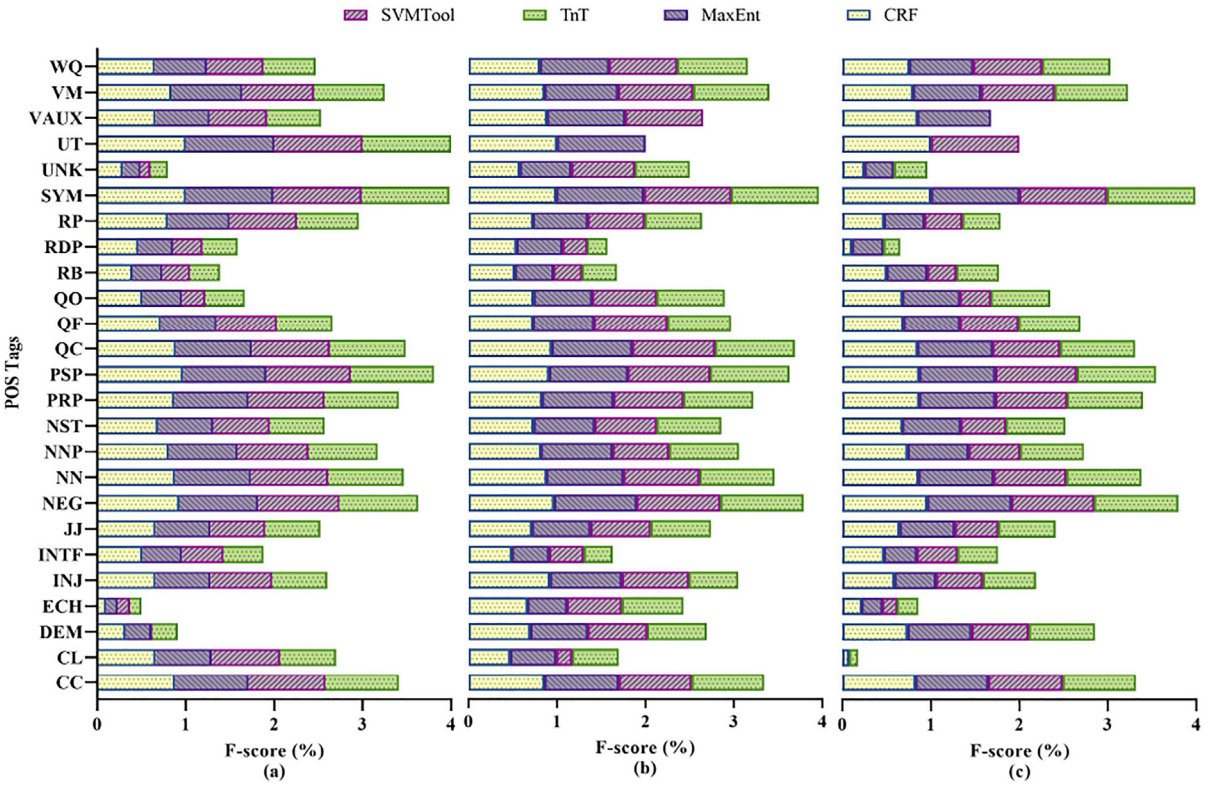

Fig. 2. F-score result of POS tagging for (a) Bhojpuri, (b) Maithili and (c) Magahi.

are the low resource languages. According to the size of the annotated data, deep learning techniques extract efficient features during training and provide more accurate results than machine learning techniques. Maithili has minimally annotated data of POS and Chunk compared to other languages for which the SAHBiLC model reported a lower result than CRF, but Fine-SAHBiLC improved SAHBiLC model performance further through passing its efficient features. The ratio of chunks (the number of tokens divides by the number of chunks) in Bhojpuri and Hindi is 1.50 and 1.85, respectively, indicating that each such word forms a Chunk. As a result, machine learning techniques provide better results compared to deep learning-based techniques. It might also perform better for morphologically complex languages, as fine-tuning might help in learning morphological idiosyncrasies.

The SAHBiLC model for Bhojpuri can accurately predict on the less frequent and most frequent tags, while Fine-SAHBiLC model performs prediction on less frequent and most frequent tags of Maithili (except UT, or utterance tag) and Magahi (except CL, UT, UNK) thoroughly, as shown in Fig. 4. Apart from this, the Fig. 5 shows that the tags with below 50% accuracy in terms of F-score have degraded performance in Bhojpuri after successfully applying Fine-SAHBiLC model. In contrast, Maithili and Magahi have attained the improvement by 50% for INJ and 30% for INJ, INTF, respectively. This may be due to the fact that Bhojpuri is close to Hindi than the other two languages.

Furthermore, the less frequent (except VGNN) and the most frequent chunk tags in Bhojpuri attained improvement after applying the SAHBiLC model. Maithili attained the increment by 50% on less

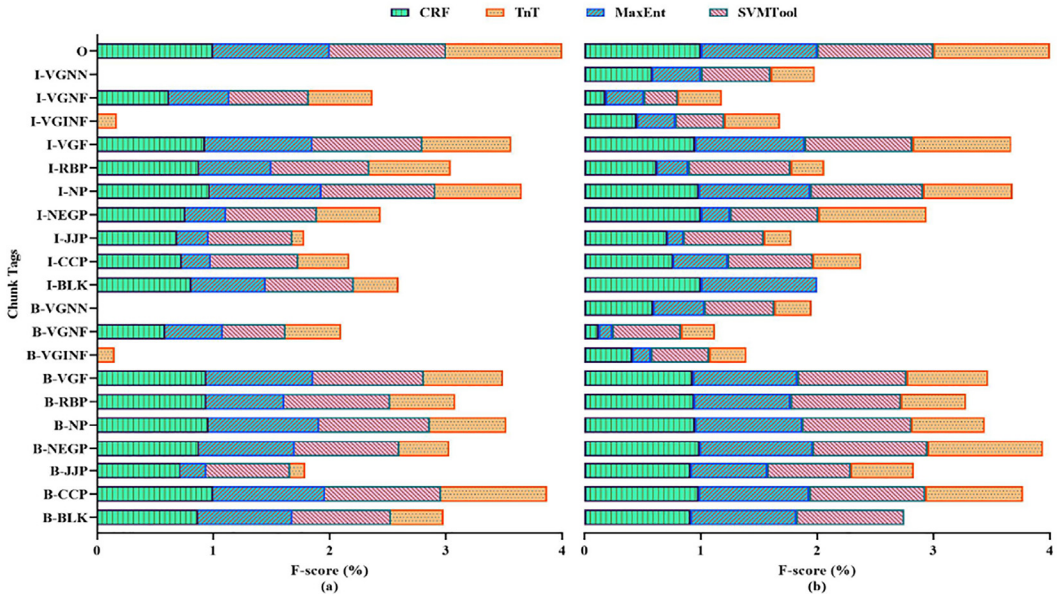

Fig. 3. F-score results of Chunk tagging for (a) Bhojpuri and (b) Maithili.

frequent and 5% on most frequent chunk tags after using Fine-SAHBiLC, as shown in Fig. 6.

The SAHBiLC on Hindi improved the result compared to LSTM-CNN-CRF and machine learning techniques for POS tagging. Whereas the Chunking result is a bit low compare to CRF.

The performance for POS tagging has indeed turned out to be better in case of deep learning techniques than traditional machine learning techniques for the rest of the languages. Bhojpuri has a higher amount of data as compared to both Maithili and Magahi. An effective solution is provided by employing data from a related task when a specific target task dataset is scarce. However, when shifting knowledge from less relevant data, it may reduce the performance of target task, is described as negative transfer. The SAHBiLC and CRF model turn out be more accurate than the Fine-SAHBiLC due to negative transfer for all POS tags on Bhojpuri. On the other hand, Fine-SAHBiLC attains better performance in POS tagging for both Maithili and Magahi. In the case of Chunking, Fine-SAHBiLC degrades the overall accuracy for Bhojpuri. Therefore, CRF is a better approach to Bhojpuri and Fine-SAHBiLC achieves better accuracy than other techniques for Maithili.

Table 6
Results for Deep Learning techniques in terms of **A**ccuracy, **P**recision, **R**ecall and **F**-score for POS Tagging.

Language	Technique	A	P	R	F
Bhojpuri	LSTM-CNN-CRF	0.84	0.84	0.84	0.84
	SAHBiLC	0.86	0.87	0.87	0.87
	Fine-SAHBiLC	0.85	0.86	0.85	0.85
Maithili	LSTM-CNN-CRF	0.83	0.82	0.82	0.82
	SAHBiLC	0.84	0.85	0.84	0.84
	Fine-SAHBiLC	0.86	0.86	0.86	0.86
Magahi	LSTM-CNN-CRF	0.83	0.83	0.84	0.83
	SAHBiLC	0.83	0.84	0.84	0.84
	Fine-SAHBiLC	0.86	0.87	0.87	0.86
Hindi	LSTM-CNN-CRF	0.92	0.94	0.91	0.92
	SAHBiLC	0.95	0.95	0.95	0.95

Table 7
Results for Deep Learning techniques in terms of **A**ccuracy, **P**recision, **R**ecall and **F**-score for Chunking.

Language	Technique	A	P	R	F
Bhojpuri	LSTM-CNN-CRF	0.91	0.84	0.86	0.85
	SAHBiLC	0.94	0.94	0.94	0.94
	Fine-SAHBiLC	0.93	0.94	0.93	0.93
Maithili	LSTM-CNN-CRF	0.91	0.86	0.87	0.87
	SAHBiLC	0.93	0.93	0.91	0.91
	Fine-SAHBiLC	0.95	0.95	0.94	0.95
Hindi	LSTM-CNN-CRF	0.97	0.96	0.95	0.95
	SAHBiLC	0.98	0.98	0.98	0.98

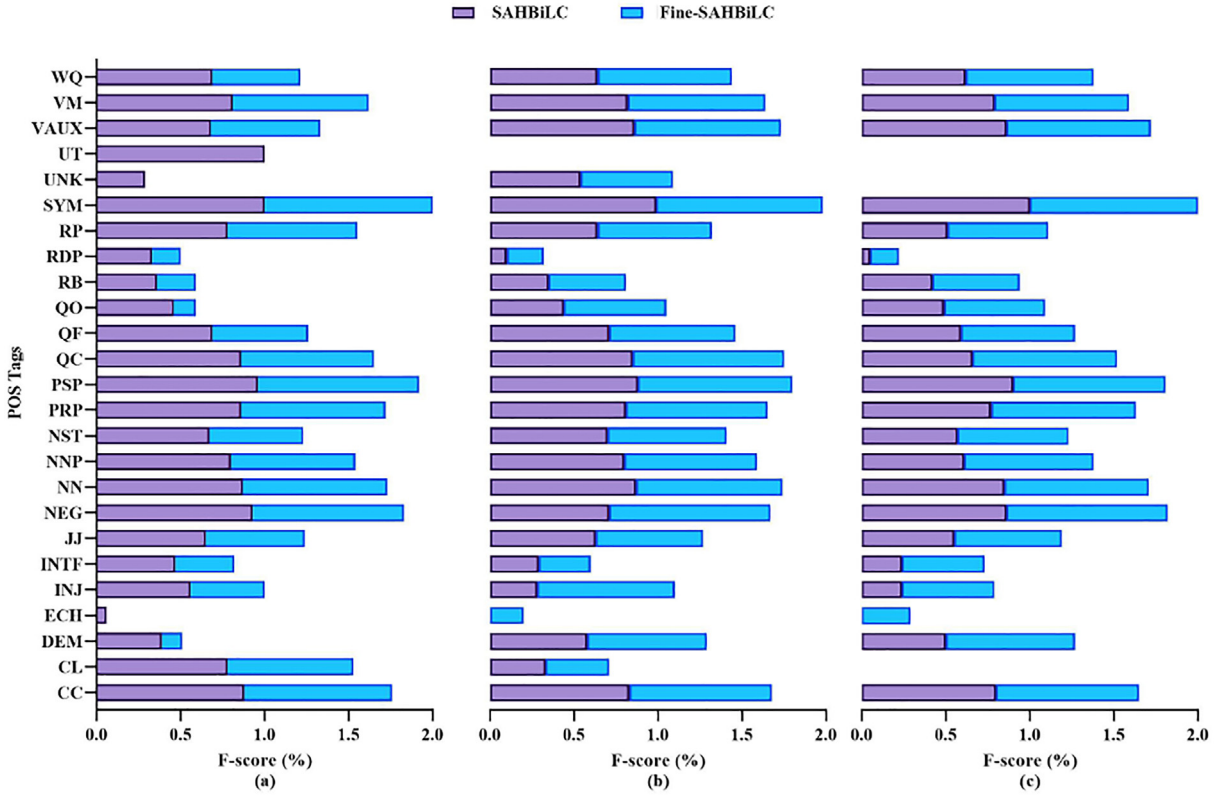

Fig. 4. F-scores of POS tagging after applying SAHBiLC model and Fine-SAHBiLC model on (a) Bhojpuri and (b) Maithili.

6. Error analysis

There are many challenges while annotating corpus of these low resourced languages, as we discussed in our paper regarding the resources used in this paper (Mundotiya et al., 2021). In spite of being labeled as dialects or varieties of Hindi, morphological constructions of Bhojpuri, Maithili and Magahi considerably differ from Hindi. Bhojpuri, Maithili and Magahi are partially synthetic languages. Thus, the use of embedded case markers, emphatic markers, classifiers, determiners etc is frequent in these languages. These linguistic idiosyncrasies and lexical ambiguity create challenges for machine learning and are responsible for many problems in annotation as well as in prediction by the algorithms.

In the next section, we discuss some such cases have. In the examples in the following section, the tags assigned by human annotator are written in ( ) brackets and tags assigned by machine are written in [ ] brackets. As some of the examples show, the machine sometimes gave correct tag even when the human annotator had errored in manual annotation, as noticed in the test data. Still, there were fewer such cases than the ones where the machine made an error.

6.1. Errors due to linguistic idiosyncrasies

6.1.1. Verb fused with negation markers

In general, negative markers occur separately in these languages but in some cases they get fused with verb. In the BIS tagset[2], we do not have a separate tag for this subcategory of 'Negative' markers. In any case, since they are functional words or morphemes used with the head word, the POS tag has given based on the head word. Sometimes this leads to faulty annotation. For example:

Bhojpuri: kavano (*PRP*)[*PRP*] bAwa (*NN*)[*NN*] naiKe (*NEG*)[*VM*]. (*SYM*)[*SYM*].

Hindi Translation: kol bAwa nahIM hE.

English Translation: It doesn't matter.

Magahi: eka (*QC*)[*QC*] azjurI (*NN*)[*NN*] ParahI-PutahA (*NN*)[*NN*] ke(*PSP*)[*PSP*] neMvAnna (*VM*)[*VM*] naz (*RP*)[*PSP*]. (*SYM*)[*SYM*].

Hindi Translation: eka azjali (BI) ParuhI-Buje xAne navAnna nahIM hEM.

English Translation: There is not even a handful of roasted puffed-rice grains of the new harvest (to eat).

In the first example from Bhojpuri, the word *naIKe* is assigned the tag NEG (negation marker) by the annotator. However, in the sentence, it functions like a verb, and the machine has correctly tagged it as VM (main verb). In the second example from Magahi, the word *naz* similarly functions as a verb (VM), but the annotator has tagged it as particle (RP), while the machine has tagged it as post-position (PSP). The forms of both the words like negation markers. In Bhojpuri they are correctly identified by the machine, even when the annotator is wrong, perhaps because there is significantly much more training data for Bhojpuri than for Magahi. Magahi is also morphological more complex. Apart from that, there is no obvious verb morpheme in the second example, whereas it is in the first example.

6.1.2. Embedded case markers

In these languages case markers are usually separate from the head word, but in many cases they get merged with nominals and pronominals. This is especially true of locative, instrumental

[2] http://tdil-dc.in/tdildcMain/articles/134692Draft%20POS%20Tag%-20standard.pdf.

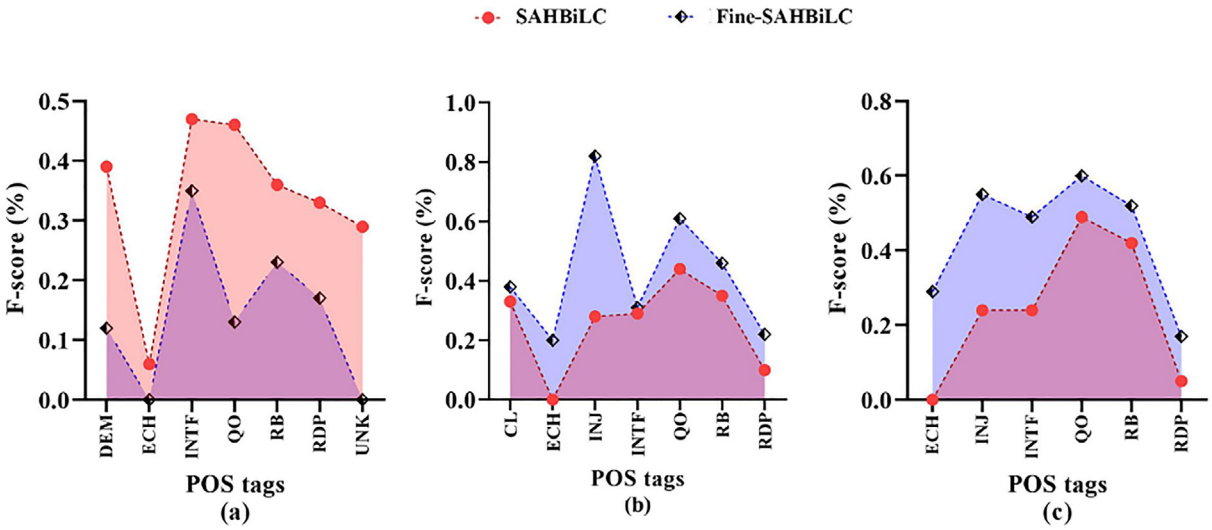

Fig. 5. Most affected POS tags after applying SAHBiLC model and Fine-SAHBiLC model on (a) Bhojpuri, (b) Maithili and (c) Magahi.

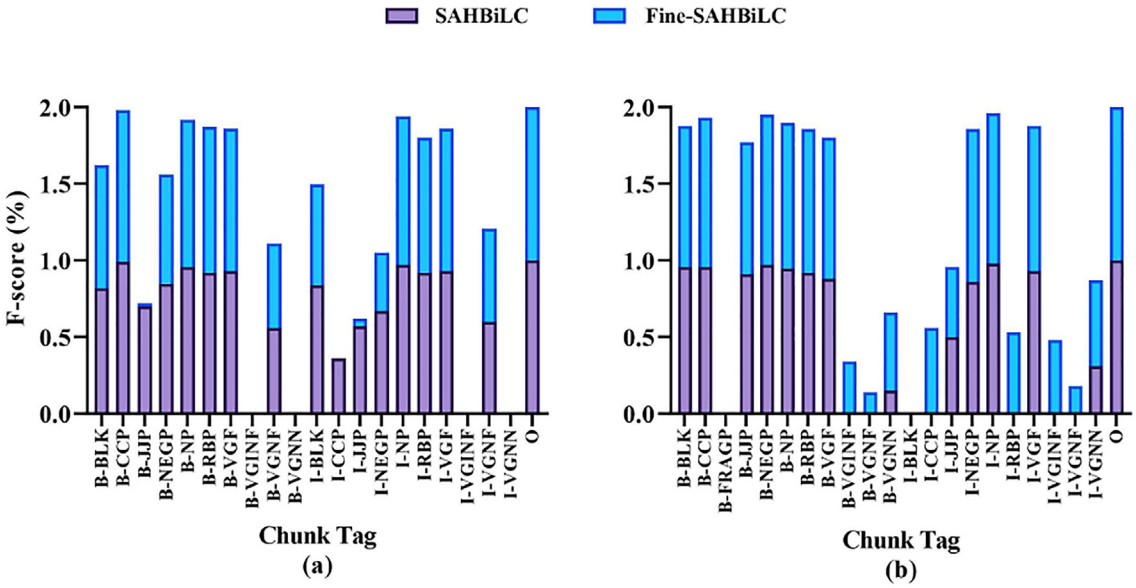

Fig. 6. The F-scores of chunk tagging after applying SAHBiLC model and Fine-SAHBiLC model on (a) Bhojpuri and (b) Maithili.

and genitive case markers. This construction feature also creates challenges for machine learning. For example:

- **Nominals**

- **Bhojpuri:** sAzJe (*NN*)[*RB*] suwa (*VM*)[*NN*] jAilAz (*VAUX*)[*VM*]. (*SYM*)
  [*SYM*]
  Hindi Translation: SAma ko hI so jAwe hEM.
  English Translation: (He/She honorific, or they, usually)[3] Fall a sleep in the evening.
- **Maithili:** o (*PRP*)[*PRP*] gAmakez (*NN*)[*NN*] ekawAka (*JJ*)[*NN*] sUwrame (*NN*)[*NN*] banhane (*VM*)[*VM*] CaWi (*VAUX*)[*VAUX*]. (*SYM*)[*SYM*]
  Hindi Translation: unhoMne gAzva ko ekawA ke sUwra meM bAzXA WA.
  English Translation: [He/She honorific, or they] tied the village with the thread of unity.
- **Magahi:** rAwe (*NST*)[*JJ*] iskUla (*NN*)[*NN*] ke (*PSP*)[*PSP*] sainaboda (*NN*)[*NN*] jagaha-jagaha (*NN*)[*NST*] se (*PSP*)[*PSP*] lataka (*VM*)[*VM*] gela (*VAUX*)[*VAUX*] hala
  (*VAUX*)[*VAUX*]. (*SYM*)[*SYM*].
  Hindi Translation: rAwa meM skUla kA sAinaborda jagaha-jagaha se lataka gayA WA.
  English Translation: The same night, the school signboard was hanging from place to place.

- **Pronominals**

- **Bhojpuri:** hamare (*PRP*)[*PSP*] rUpa (*NN*)
  [*NN*] U (*DEM*)[*PRP*] cAroM (*QO*)[*QC*] orI (*NST*).
  [*NST*] nihArelI (*VM*)[*NN*] .(*SYM*)[*SYM*].
  Hindi Translation: mere rUpa ko vaha cAroM ora se nihArawI hE.
  English Translation: She looks at me from all four sides.

[3] *Note*: In English translations, ellipsis is denoted with parenthesis, whereas alternative word translations are denoted by square brackets.

In case of nominals, there is inconsistency in annotation of Bhojpuri and Magahi sentences. While the words *sAzJe* (in the evening) and *rAwe* (in the night) are tagged by annotators as NN (noun) and NST (relational noun), respectively. The machine tags the first as adverb (RB) and second as adjective (JJ), which are easy mistakes to make in the absence of enough data. The convention with BIS tagset is to tag such words as NST.

6.1.3. Diverse realizations of a single token

Analogous to other Indian languages Bhojpuri, Maithili and Magahi languages also have several cases of diverse realization of a single token. These tokens generally have multiple functional and connotative meanings. In the annotated data of Bhojpuri language the token *t* has the highest 9 tags for different realizations. Similarly in Magahi, 9 tags are assigned to the token *khaali*, whereas in Maithili the token *lel* has been assigned 10 tags for diverse realizations. List of such tokens is long in all these three languages.

6.1.4. Homophonous forms

Bhojpuri, Maithili and Magahi also have homophonous words in abundance, like many other Indian languages. These words look similar but their POS tags are varied which may cause confusion for the annotation task, especially for machine learning. For example:

Bhojpuri: Baila (*VM*)[*VM*] biAha (*NN*)[*NST*] mora (*PRP*)[*PRP*] karaba (*VM*)[*VM*] kA (*WQ*)[*PSP*] ?.
(*SYM*)[*SYM*]

Hindi Translation: ho gayA vivAha aba karUz kyA ?.

English Translation: I got married, now what? sonala (*NNP*)[*NNP*] sarakArI (*NN*)[*NNP*]gavAha (*NN*)[*NN*] bana (*VM*)[*VM*] gaiIl (*VM*)[*VAUX*] .(*SYM*)[*SYM*]

Hindi Translation: sonala sarakArI gavAha bana gaI.

English Translation: Sonal became a witness from government side.

AnhIM (*NN*)[*NN*] me (*PSP*)[*PSP*] Aam (*NN*)[*JJ*] niyara (*PRP*)[*PRP*] Cappara (*NN*)[*NN*] cuawA (*VM*).
[*NN*] .(*SYM*)[*SYM*].

Hindi Translation: AzXI meM Ama ke samAna Cappara cU rahA hE.

English Translation: The shed is dripping like mangoes in the storm.

In these cases PSP, VM and JJ are the most frequent tags respectively for *kA*, *bana* and *Ama*, which led to faulty annotation by the machine. These are some examples of confusion created by homophonous words for automatic annotation.

6.1.5. Classifiers

Similar to Bengali and Oriya, the Bhojpuri, Magahi and Maithili languages also very often use classifiers with numerals, whereas Hindi and its other 'dialects' (or 'sub-languages') are classifier-less languages. Classifier markers in these languages are: *To*, *Te*, *go* and *Ke* etc. This feature often creates problems for machine learning. For example:

Bhojpuri: wIna (*QC*)[*QC*] cAra (*QC*)[*QC*] go (*CL*)[*RP*] Gara (*NN*)[*NN*] A (*CC*)[*CC*] xalAna (*NN*)[*NN*] banAvala (*VM*)[*VM*] rahe (*VM*)[*VAUX*] .(*SYM*).
[*SYM*]

Hindi Translation: wIna cAra Gara Ora xalAna banAe hue We.

English Translation: Three or four houses and verandahs were built.

Maithili: eka (*QC*)[*RP*] tA (*CL*)[*RP*] cunAva (*JJ*)[*NN*] karmacArIka (*NN*)[*NN*] niXana (*NN*)[*VM*] seho (*RB*).
[*RP*] BaZ (*VM*)[*VM*] gelanhi (*VAUX*)[*VAUX*]. (*SYM*).
[*SYM*]

Hindi Translation: eka cunAva karmacArI kA niXana BI ho gayA.

English Translation: An election worker also died.

In these example sentences machine assigned incorrect tag [RP] to the classifiers of these languages, whereas the CL tag assigned by human annotators is correct. It is an easy to mistake these classifiers for particles.

6.1.6. Amalgamated emphatic expressions

Like other languages of the Magadhi group, Bhojpuri, Maithili and Magahi languages also have the feature of merged emphatic particles with nominals and pronominals in general. This kind of idiosyncrasy, combined with lack of data, makes annotation task difficult for machine as well. For example:

Bhojpuri: u (*PRP*)[*PRP*] abbe (*NST*)[*RP*] Gare (*NN*).
[*NN*] Aila (*VM*)[*VM*] bA (*VAUX*)[*VAUX*]. (*SYM*).
[*SYM*]

Hindi Translation: vo aBi hI Gara AyA hE.

English Translation: He came home just now.

Maithili: hamahuz (*PRP*)[*RP*] wapAkasaz (*RB*)[*RB*] haz (*NN*)[*RP*] kahi (*VM*).
[*VM*] xelahuz (*VAUX*)[*VAUX*] .(*SYM*)[*SYM*].

Hindi Translation: mEMne BI wapAka se hAz kaha xiyA.

English Translation: I also said yes promptly.

Magahi: aisahIM (*DEM*)[*RB*] wo (*RP*)[*PSP*] bahanoI (*NN*)[*NN*] rusala (*VM*).
[*VM*] haWina (*VAUX*)[*VAUX*] . (*SYM*)[*SYM*].

Hindi Translation: Ese hI wo bahanoI rUTe We.

English Translation: Brother-in-law was angry without any reason.

In Bhojpuri, the emphasized temporal adverb *abbe* is tagged by the annotator as a relational noun (NST), while the machine mistakes it for a particles (RP). This indicates there are still problems with the annotated data. In Maithili, the emphasized pronoun *hamahuz* is correctly tagged by the annotator as pronoun (PRP), but the machine tags it as a particle (RP), which is perplexing. In Magahi, the demonstrative *aisahIM* is wrongly tagged by the machine as particle again. It seems that due to the a frequent use of particles in these languages, words are often wrongly identified as particles.

6.1.7. Interrogative markers

The machine frequently assigned wrong tags to different interrogative markers of Bhojpuri, Maithili and Magahi. This may be because most of these markers are homophonous forms. For example:

Bhojpuri: I (*DEM*)[*PRP*] BojapurI (*NNP*)[*JJ*] pawrikA (*NN*)[*NN*] ke (*PSP*)[*PSP*] kaise (*WQ*)[*PRP*] paDZaba (*VM*)[*NN*] ? (*SYM*)[*SYM*].

Hindi Translation: isa BojapurI pawrikA ko kEse paDUz ?.

English Translation: How do I read this Bhojpuri magazine?.

Magahi: xeKahIM (*VM*)[*VM*] wa (*CC*)[*RP*] ke (*WQ*)[*RP*] hai (*AUX*)[*AUX*] xuariyA (*NN*)[*NN*] para (*PSP*)[*PSP*] . (*SYM*)[*SYM*].

Hindi Translation: xeKUz wo kOna hE xaravAje para.

English Translation: Let me see who is at the door.

Maithili: bahasa (*NN*)[*NN*] para (*PSP*)[*PSP*] kaya (*WQ*)[*PSP*] tA (*RP*)[*RP*] kameMta (*NN*)[*NN*] Ayala (*VM*)[*VM*] aCi (*VAUX*)[*VAUX*] ? (*SYM*)[*SYM*].

Hindi Translation: bahasa para kiwane kameMta Ae hEM ?.

English Translation: How many comments have come on the debate ?.

7. Conclusion

Sequential labelling is one of the preliminary tasks for processing for any new languages in NLP. Most of the diverse Indian languages are low resourced, which restricting the development of

human language technology. In this work, we have proposed two deep learning models, namely SAHBiLC (monolingual embedding) and Fine-SAHBiLC (character-level transfer learning). The obtained results have been compared among the proposed deep learning models, state-of-the-art deep learning model and traditional machine learning techniques (TnT, CRF, MaxEnt, SVMTool) for POS tagging on Bhojpuri, Maithili and Magahi, and Chunking on Bhojpuri, Maithili annotated corpus. The designing of an annotated corpus for POS tagging, Chunking, and building an automatic tool for these tasks is the first such attempt towards all these three languages. The SAHBiLC and Fine-SAHBiLC outperform on Bhojpuri and Maithili, Magahi, respectively for both tasks. This indicates that fine-tuning is helpful in case of less training data and for complex morphology. In future work, we can incorporate the available pre-trained multi-lingual embeddings of similar languages to counter the negative transfer while applying transfer learning for both the tasks. Further, we can enhance the performance of deep learning models by using semi-supervised techniques.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Bullinaria, J.A., Levy, J.P., 2012. Extracting semantic representations from word co-occurrence statistics: stop-lists, stemming, and svd. Behav. Res. Methods 44, 890–907.

Christianson, C., Duncan, J., Onyshkevych, B., 2018. Overview of the darpa lorelei program. Mach. Transl. 32, 3–9.

Collobert, R., Weston, J., 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In: Proceedings of the 25th international conference on Machine learning, ACM, pp. 160–167.

Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., Harshman, R., 1990. Indexing by latent semantic analysis. J. Am. Soc. Inf. Sci. 41, 391–407.

Dozat, T., Qi, P., Manning, C.D., 2017. Stanford's graph-based neural dependency parser at the CoNLL 2017 shared task. In: Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies, Association for Computational Linguistics, Vancouver, Canada. pp. 20–30. URL: https://www.aclweb.org/anthology/K17-3002, doi:10.18653/v1/K17-3002.

Huang, Z., Xu, W., Yu, K., 2015. Bidirectional lstm-crf models for sequence tagging. arXiv preprint arXiv:1508.01991.

Kann, K., Bjerva, J., Augenstein, I., Plank, B., Søgaard, A., 2018. Character-level supervision for low-resource pos tagging. In: Proceedings of the Workshop on Deep Learning Approaches for Low-Resource NLP. pp. 1–11.

Kim, J.K., Kim, Y.B., Sarikaya, R., Fosler-Lussier, E., 2017. Cross-lingual transfer learning for pos tagging without cross-lingual resources. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pp. 2832–2838.

Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., Dyer, C., 2016. Neural architectures for named entity recognition. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, San Diego, California. pp. 260–270. URL: https://www.aclweb.org/anthology/N16-1030. doi:10.18653/v1/N16-1030.

Levy, O., Goldberg, Y., 2014. Neural word embedding as implicit matrix factorization. Advances in neural information processing systems, 2177–2185.

Ma, X., Hovy, E.H., 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7–12, 2016, Berlin, Germany, Volume 1: Long Papers, The Association for Computer Linguistics. doi:10.18653/v1/p16-1101.

Mishra, P., Mujadia, V., Sharma, D.M., 2017. Pos tagging for resource poor languages through feature projection. In: Proceedings of the 14th International Conference on Natural Language Processing (ICON-2017). pp. 50–55.

Mundotiya, R.K., Singh, M.K., Kapur, R., Mishra, S. and Singh, A.K., 2021. Linguistic Resources for Bhojpuri, Magahi, and Maithili: Statistics about Them, Their Similarity Estimates, and Baselines for Three Applications. Transactions on Asian and Low-Resource Language Information Processing, 20 (6),1-37. doi: https://doi.org/10.1145/3458250.

Murthy, R., Khapra, M.M., Bhattacharyya, P., 2018. Improving ner tagging performance in low-resource languages via multilingual learning. ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP) 18, 9.

Plank, B., Søgaard, A., Goldberg, Y., 2016. Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). Association for Computational Linguistics, Berlin, Germany, pp. 412–418. https://doi.org/10.18653/v1/P16-2067. URL: https://www.aclweb.org/anthology/P16-2067.

Priyadarshi, A., Saha, S.K., 2020. Towards the first maithili part of speech tagger: resource creation and system development. Comput. Speech Language 62, 101054.

PVS, A., Karthik, G., 2007. Part-of-speech tagging and chunking using conditional random fields and transformation based learning. Shallow Parsing for South Asian Languages 21.

Saha, S.K., Sarkar, S., Mitra, P., 2008. A hybrid feature set based maximum entropy hindi named entity recognition. In: Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-I.

Santos, C.D., Zadrozny, B., 2014. Learning character-level representations for part-of-speech tagging. In: Proceedings of the 31st International Conference on Machine Learning (ICML-14). pp. 1818–1826.

dos Santos, C.N., Zadrozny, B., 2014. Learning character-level representations for part-of-speech tagging. In: Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014, JMLR.org. pp. 1818–1826. URL: http://proceedings.mlr.press/v32/santos14.html.

Singh, T.D., Ekbal, A., Bandyopadhyay, S., 2008. Manipuri pos tagging using crf and svm: a language independent approach. In: proceeding of 6th International conference on Natural Language Processing (ICON-2008). pp. 240–245.

Subbārāo, K.V., 2012. South Asian languages: A syntactic typology. Cambridge University Press.

Tandon, J., Chaudhry, H., Bhat, R.A., Sharma, D., 2016. Conversion from paninian karakas to universal dependencies for Hindi dependency treebank. In: Proceedings of the 10th Linguistic Annotation Workshop held in conjunction with ACL 2016 (LAW-X 2016), Association for Computational Linguistics, Berlin, Germany. pp. 141–150. doi:10.18653/v1/W16-1716.

Turian, J., Ratinov, L., Bengio, Y., 2010. Word representations: a simple and general method for semi-supervised learning. In: Proceedings of the 48th annual meeting of the association for computational linguistics. Association for Computational Linguistics, pp. 384–394.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need. Advances in neural information processing systems, 5998–6008.

Yang, Z., Salakhutdinov, R., Cohen, W.W., 2017. Transfer learning for sequence tagging with hierarchical recurrent networks. In: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24–26, 2017, Conference Track Proceedings, OpenReview.net. URL: https://openreview.net/forum?id=ByxpMd9lx.

Zhang, Y., Chen, H., Zhao, Y., Liu, Q., Yin, D., 2018. Learning tag dependencies for sequence tagging. In: IJCAI. pp. 4581–4587.