Journal of King Saud University – Computer and Information Sciences 34 (2022) 7324–7334

Contents lists available at ScienceDirect

Journal of King Saud University – Computer and Information Sciences

journal homepage: www.sciencedirect.com

Integration of morphological features and contextual weightage using monotonic chunk attention for part of speech tagging

Rajesh Kumar Mundotiya [a,*], Arpit Mehta [a], Rupjyoti Baruah [a], Anil Kumar Singh [a]

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

ARTICLE INFO

Article history:
Received 4 July 2021
Revised 17 August 2021
Accepted 18 August 2021
Available online 27 August 2021

Keywords:
Part of Speech tagging
Morphological features
Attention mechanism
Convolutional neural network

ABSTRACT

Part-of-Speech (POS) tagging is a fundamental sequence labeling problem in Natural Language Processing. Recent deep learning sequential models combine the forward and backward word informatio for POS tagging. The information of contextual words to the current word play a vital role in capturing the non-continuous relationship. We have proposed Monotonic chunk-wise attention with CNN-GRU-Softmax (MCCGS), a deep learning architecture that adheres to these essential information. This architecture consists of Input Encoder (IE), encodes word and character-level, Contextual Encoder (CE), assigns the weightage to adjacent word and Disambiguator (D), which resolves intra-label dependencies as core components. Moreover, different morphological features have been integrated into the core components of MCCGS architecture as MCCGS-IE, MCCGS-CE and MCCGS-D. The MCCGS architecture is validated on the 21 languages from Universal Dependency (UD) treebank. The state-of-the-art models, Type constraints, Retrofitting, Distant Supervision from Disparate Sources and Position-aware Self Attention, MCCGS and its variants such as MCCGS-IE, MCCGS-CE and MCCGS-D are obtained mean accuracy 83.65%, 81.29%, 84.10%, 90.18%, 90.40%, 91.40%, 90.90%, 92.30%, respectively. The proposed model architecture provides state-of-the-art accuracy on the low resource languages as Marathi (93.58%), Tamil (87.50%), Telugu (96.69%) and Sanskrit (97.28%) from UD treebank and Hindi (95.64%) and Urdu (87.47%) from Hindi-Urdu multi-representational treebank.

1. Introduction

Natural Language Processing (NLP) is an application of Artificial Intelligence, which comprises text and speech processing-based applications. Part of Speech (POS) tagging is a preliminary task for text processing to assign a grammatical category to a word in a sentence. It plays a vital role in various NLP applications such as syntactic parsing, word sense disambiguation, machine translation system, question answering, sentiment analysis, co-reference resolution, text classification, social media content classification, and natural language understanding.

A 'word' in a text carries linguistic information as lexical categories such as noun, verb etc., and grammatical features such as gender, number, person etc. The fine-grained POS categories provide a linguistic clue (syntactic information) to decide the appropriate POS category of a word within a sentence or phrase. Apart from that, semantic information is also encoded within a word or sentence. POS categories can be used to disambiguate the multiple answers provided by a morphological analyzer. The morphological analyzer captures semantic information in a conventional machine translation systems. Machine translation system translates every word from a source language to a target language. Yin et al. (2019) observed that the Proper Noun POS category word had not been translated correctly in the European languages. They tried to overcome this problem by jointly modeled with neural machine translation and POS tagging using Multi-Task learning. A piece of syntactic information obtained from the POS data leads to a better encoding of source-sentence structure's during the generation from the machine translation system (Niehues and Cho, 2017).

With conventional machine learning techniques for POS tagging, there was a heavy reliance on feature engineering such as prefixes, suffixes, contextual words, and language-specific features, i.e., capitalization and creating several handcrafted features. For

* Corresponding author.
E-mail addresses: rajeshkm.rs.cse16@iitbhu.ac.in (R.K. Mundotiya), arpitmehta.cse18@iitbhu.ac.in (A. Mehta), rupjyotibaruah.rs.cse18@iitbhu.ac.in (R. Baruah), aksingh.cse@iitbhu.ac.in (A.K. Singh).
Peer review under responsibility of King Saud University.

Production and hosting by Elsevier

https://doi.org/10.1016/j.jksuci.2021.08.023
1319-1578/

example, Conditional Random Fields (CRF) and Maximum Entropy (ME) work on symbolic features that include lexicon, affixes and other morphological information to improve POS tagging performance. The availability of lexicons and basic morphological information is facile, while dependency information is unfortunate due to profound linguistic knowledge. These features are directly incorporated into the machine learning algorithms. However, deep learning includes these features as dense representation in the model. Dense feature representation is an adequate representation that learned itself from the provided value of the feature.

Even integrating these dense features in deep learning models cannot effectively address the long-range non-continuous relation dependencies. The non-continuous dependency defined by word depends on its semantic information and depends upon the information of its contextual neighbor; hence, it plays a significant role in POS tagging. This phenomenon has explained in Fig. 1, where the word "communities" is dependent upon the word "it", as resultant it tagged with the "NNP" (singular proper noun). If the non-continuous dependency ignored here, the tag could be "NNPS" (plural proper noun) due to suffix information of '-s' or '-es'. Similarly, the morphological information provides clues about the category of a word viz. the word "communities" consists of the "Nom" (nominative) case and person information "Person" as 3rd. This information usually exhibits for the noun categories.

Previously proposed approaches are based on either attention-based contextual information (Lin et al., 2021; Shao et al., 2021; Mundotiya et al., 2020) or hand-crafted features (existing linguistic knowledge) (Plank and Agić, 2018; Plank and Klerke, 2019; Chakrabarty et al., 2019) for non-continuous relation modeling but have not brought together as needed, which improves the POS tagging performance. Apart from this, there is less profound study on the feature inclusion that at which layer integration improves the performance of POS tagging as deep learning follows a layered approach. Hence, in this paper, attention mechanism and inclusion of morphological features have been employed in a novel neural architecture, Monotonic Chunk-wise attention with CNN-GRU-SOFTMAX (MCCGS) comprising three essential components. The first component, input encoder, is responsible for encoding intra-word information using character and word embeddings, where word embedding from characters obtained through CNN. The dependencies among words encoded by bidirectional GRU and assign weights as contextual features to the words using monotonic chunk-wise attention as the second component, denoted as a contextual encoder in this paper. The third component, disambiguator, resolves the label dependencies using bidirectional GRU and decodes the labels' probability distribution using softmax. Furthermore, in the proposed MCCGS architecture, the dense representation of handcrafted features (morphological features) at all three components have been performed. All these extensive experiments have been performed on the 21 languages from the Universal Dependency treebank dataset. The summary of silent contributions of this paper are as follows:

- A novel neural POS tagging approach proposed, assigning high weightage to adjacent fixed words to the current word by using a monotonic chunk-wise attention mechanism with the dense representation of morphological features.
- Provides state-of-the-art empirical results after incorporating morphological features at different layers to the proposed approach on Universal Dependency treebank.
- The robustness of the approach was evaluated on two different datasets of Indo-Aryan languages. The first dataset contains the Hindi and Urdu languages belonging to the Hindi-Urdu multi-representative treebank. A treebank with very little annotated data in Universal Dependency treebank has been selected in the second dataset, comprising Tamil, Telugu, Marathi and Sanskrit languages. We have obtained significant improvements in comparison to existing state-of-the-art results on both datasets.

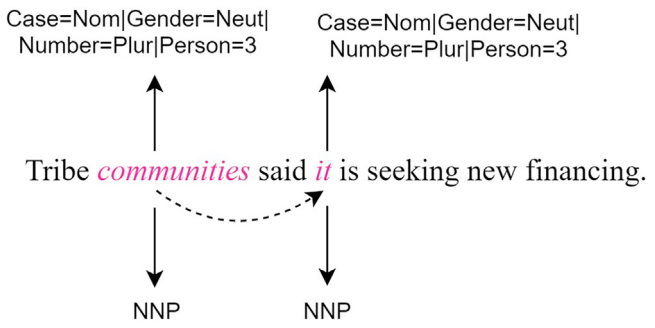

Fig. 1. Impact of morphological features and longer contextual dependencies.

2. Related work

Traditionally, most of the high score POS tagging approaches were based on the probabilistic and statistical learning algorithms such as Hidden Markov Model (Kupiec, 1992), Maximum Entropy Markov Model (Ratnaparkhi, 1996), Conditional Random Fields (Lafferty et al., 2001), Semi Markov Random Field (semi-CRF) (Sarawagi and Cohen, 2004), Support Vector Machine (Kudo and Matsumoto, 2001), Hidden Markov Support Vector Machine (Altun et al., 2003), required well designed hand-crafted features or language specific features during training in a supervised manner.

Following the availability of ample data, a deep learning model has emerged that does not rely on traditional features and is surpasses probabilistic and statistical learning algorithms for most of the text processing tasks, including POS tagging. Applying neural networks (Lu et al., 2003; Hinton et al., 2006) to the problem of POS tagging on the vast amount of annotated data is not a new research topic. Santos et al. (2014) used Convolutional Neural Network (CNN) for word representation through character level for POS tagging.

Although, these approaches neglect the global long-range dependencies among the words of the sentences; therefore, variants of Recurrent Neural Network (RNN), i.e. Gated Recurrent Unit (GRU), Long Short Term Memory (LSTM), Bidirectional-LSTM (Bi-LSTM) and Bidirectional-GRU (Bi-GRU) proposed to capture features of longer dependencies for each word, and consequently achieved good performance. At the very beginning, Huang et al. (2015) used the LSTM unit as a word information encoder with CRF as a label sequence decoder. Later on, Ling et al. (2015) proposed a Character to Word (C2W) model that generates word embedding by considering characters as an atomic unit using Bi-LSTM, which is further encoded by Bi-LSTM followed by softmax to decode the labels. Ma and Hovy (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 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.

Liu et al. (2018) used all characters of sentences along with word boundaries for generating word embedding through Bi-LSTM, which was further employed with highway network and Bi-LSTM in a parallel fashion to decode label dependencies by CRF. 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.

The majority of earlier proposed work on neural POS tagging assumed that handcrafted features antiquated for deep learning-based models and uniquely depended on an end to end training. However, Faruqui et al. (2015) earlier work combines semantic symbolic features with word embedding. Similarly, Sagot and Alonso (2017) use morphological lexicons as additional input, collected from Apertium and Alexina lexicons (language-specific) as n–hot features in the Plank et al. (2016) model, which uses Bi-LSTM for input encoder and CRF for label decoder.

The impact on the deep learning model by using handcrafted features provides a significant gain in performance. Recently, the research communities have been trying to make a robust model by improving self feature learning through the attention mechanism on the assumption of learning of the contextual features for the POS tagging. The attention mechanism helps to capture the non-continuous relationship among the words of a sentence. Mundotiya et al. (2020) have proposed self-attention and monotonic chunk-wise attention-based model and experimented on the Hindi dataset to handling non-continuous relationship within a sentence and a separate window by using a respective attention mechanism. Similarly, Wei et al. (2021) proposed a new model based on the attention mechanism. In this paper, standard additive self-attention and position-aware self-attention mechanism exploited to implicitly encode positional information at discrete and variable-length of an input sequence, respectively, to provide complementary context information based on Bi-LSTM a global sequence encoder. Shao et al. (2021) used self-attention at the word and sentence level to obtain the contextual information on the same global sequence encoder, which further used for disambiguating the labels of words by the semi-CRF approach.

Table 1 refers to a comparative overview of different deep learning models for POS tagging correlated to our approach. It depicts the input encoder with CNN, RNN and pre-trained embeddings, features comparison with contextual and handcrafted and decoder with CRF and softmax resembling with remarkable recent systems.

3. Monotonic Chunk-wise attention with CNN-GRU-Softmax (MCCGS) Architecture

To design the MCCGS model architecture, we have followed Ma and Hovy (2016), in which a character and word information has fused into the deep learning-based model before the core component of the model, LSTM or GRU. This model has extended by the attention mechanism in our proposed deep learning architecture. Recently, attention mechanisms have been gaining success in speech, image and textual processing. For the textual processing in the neural machine translation system, the attention precisely aligns the source and target words in the pair. This alignment considers the contextual information of input with their non-continuous relationship and assigns weight to themselves for the next word. Here, we leveraged the attention's advantage, i.e., contextual information and non-continuous relationship, in an antagonistic measure for POS tagging. Input encoder, contextual encoder, and disambiguator are the essential core components of the proposed MCCGS architecture, as illustrated in Fig. 2.

3.1. Input encoder

Let $D = \{(x^j, y^j)|1 \leqslant j \leqslant N\}$is a labeled sentence. Here, D is a single training sentence which belongs the training data $X \in \{D_1, D_2, \ldots, D_m\}$. The x^j denotes the word and y^j denotes the corresponding POS label of sentence D. The POS label y^j belongs to the q labels, which are represented as $y^j \in \{y_1, y_2, \ldots, y_q\}$.

In this component, MCCGS takes the given input sentence D to obtain the vector representation of each word x^j by learnable word embedding and character level embedding. The learnable word embedding captures syntax and semantic information of the word which is enhanced through the adherence of morphological information by character level embedding.

The learnable word embedding has obtained from the one-hot vector representation at the size of unique word vocabulary. All the sentences in X should have length N. If the length of D bigger than N, the antecedent word will be removed and padding will be applied if the attribute is smaller than N.

This representation and random vector W_x of embedding size, which is trainable, exhibit latent vector. Such exhibited latent vectors for all words $x^{1:N}$ in a sentence D, attained after passing to a fully connected layer with deactivated bias have been considered word embedding $v^{1:N}$.

$$v^{1:N} = W_x.x^{1:N} \tag{1}$$

For instance, the sentence (Fig. 1) "Tribe communities said it is seeking new financing" considered as input is a sequence of words. Each unique word represents a number in word encoding, and the number becomes a random generated real-valued vector. These random generated random vectors update themselves over the model learning.

For obtaining the character level word embedding, let a word x^j has $\mathscr{C}$ characters, where $\mathscr{C} = \{c^1, c^2, \ldots, c^i|1 \leqslant i \leqslant k\}$. The *character lookup dictionary* maps character identity with its one-hot vector representation. For each character, c^i in x^j is denoted by a one-hot vector representation, which has an equal size of k, using the padding operation. The padding operation appends the special symbol $< pad >$ at the end of the word that padding has applied

Table 1
Comparative overview of deep learning models for POS tagging. Here, RNN and CRF refer to their variants such as LSTM, GRU, Bi-LSTM and Bi-GRU, and semi-CRF. Embedding refers to pretrained embedding.

Systems	Input Encoder			Features		Decoder	
	CNN	RNN	Embedding	Contextual	Handcrafted	CRF	Softmax
Huang et al. (2015)			✓			✓	
Ling et al. (2015)		✓				✓	
Ma and Hovy (2016)	✓		✓			✓	
Dozat et al. (2017)		✓	✓				✓
Plank et al. (2016)		✓	✓				✓
Sagot and Alonso (2017)		✓	✓		✓		✓
Plank and Klerke (2019)		✓	✓		✓		✓
Mundotiya et al. (2020)	✓			✓		✓	
Wei et al. (2021)		✓	✓	✓		✓	✓
Lin et al. (2021)		✓	✓	✓		✓	

on the same at the sentence level. If the length of x^j is greater than k, initial characters $c^{i:k}$ have considered. Now convolution, the first operation of CNN, has been applied with filter F. Let z is the number of filters used with f kernel sizes. The filters move over the region of x^j and generate a set of features C, referred to as *feature maps* which are $\{r_1, r_2, \ldots, r_l\} \in C$. The l is calculated on the basis of kernel size and word length, $l = k - f + 1$.

$$r_i = \phi(F.c_{i:i+k-1} + b_c) \tag{2}$$

Here, F, b_c and ϕ are filter, bias and non-linear function, respectively. A piece of relevant information from all features has leveraged maximum pooling, another building block of a CNN.

$$C_{\max} = \max\{r_1, r_2, \ldots, r_l\} \tag{3}$$

$$C = \left\{C_{\max_1}, C_{\max_2} \ldots, C_{\max_z}\right\} \tag{4}$$

The generated univariant vectors from all features have concatenated, C and have passed to three stacked fully connected layers. The resulted vector, p^j from the penultimate layer has character level information for the word.

$$p^j = \phi(W_C.C + b_C) \tag{5}$$

The W_C and b_C are learning parameters and ϕ is a *ReLU* nonlinear function. This character level word vector generation process has illustrated in Fig. 3. Here, 'thing' is considered as an input word. A padding operation ($< pad >$) has been performed, as the length of the input word is smaller than the desired. All the unique characters of a language dataset were mapped with their index positions. These indexes are character identity which is further used in *character lookup dictionary* to emit a one-hot vector on which CNN operations have been performed and generate character level embeddings. For example, convolution operation generates multiple n-grams, such as thi, hin, ing, ngs, gs$<$ *pad* $>$ based on the filter size (say 3) that is used in feature maps. Over these generated filters, pooling operation extracts relevant information passed through the fully connected layer to generate a final vector. This process is iterated over each word which generates $p^{1:N}$.

The word-level embeddings $v^{1:N}$ and character-level embeddings $p^{1:N}$ have concatenated component-wise to generate the word vector representations $w^{1:N}$. This generated word vector representation leverages morphological, syntaxial and semantical information at the word and sentence level.

$$w^{1:N} = \left[p^{1:N}; v^{1:N}\right] \tag{6}$$

3.2. Contextual encoder

The Context Encoder works in two steps to generate its output. The first step assumes an identical magnitude of words of a sentence to capture sequential information by GRU, a variant of RNN. It allows holding information of a longer timestamp. Although natural language's sentences are longer and have dependencies among words; hence Bi-GRU has been considered.

The Bi-GRU takes the word vector and produces a hidden vector for each direction. The hidden vector h^{i-1} with the word vector w^i decides which information will be forward to processing of the succeeding timestamp with a degree of relevance. In each of the timestamps forward and backward process generates hidden states, i.e., $\overrightarrow{h}^{1:N}$ and $\overleftarrow{h}^{1:N}$. Forwards $\overrightarrow{h}^{1:N}$ and backwards $\overleftarrow{h}^{1:N}$ hidden vectors have concatenated according to component-wise, that generates a new hidden vector $h^{1:N}$ as resultant.

$$\overrightarrow{h}^{1:N} = \overrightarrow{GRU}(w^1, w^2, \ldots, w^N) \tag{7}$$

$$\overleftarrow{h}^{1:N} = \overleftarrow{GRU}(w^1, w^2, \ldots, w^N) \tag{8}$$

$$h^{1:N} = \left[\overrightarrow{h}^{1:N}; \overleftarrow{h}^{1:N}\right] \tag{9}$$

The second step of the contextual encoder is an attention mechanism. Attention mechanism, which was first introduced for text-based application (Bahdanau et al., 2015), focuses on the given input's cogent part to yield a better decision. Numerous tasks based on deep learning, such as image recognition (Zheng et al., 2017), machine translation (Zheng et al., 2017; Vaswani et al., 2017; Devlin et al., 2019; Bahdanau et al., 2015; Indurthi et al., 2019; Chiu and Raffel, 2018) and text classification (Sinha et al., 2018; Sun and Lu, 2020), have significantly improved their performance after the attention mechanism is deployed. Monotonic chunk-wise attention mechanism explicitly leverages the flexible

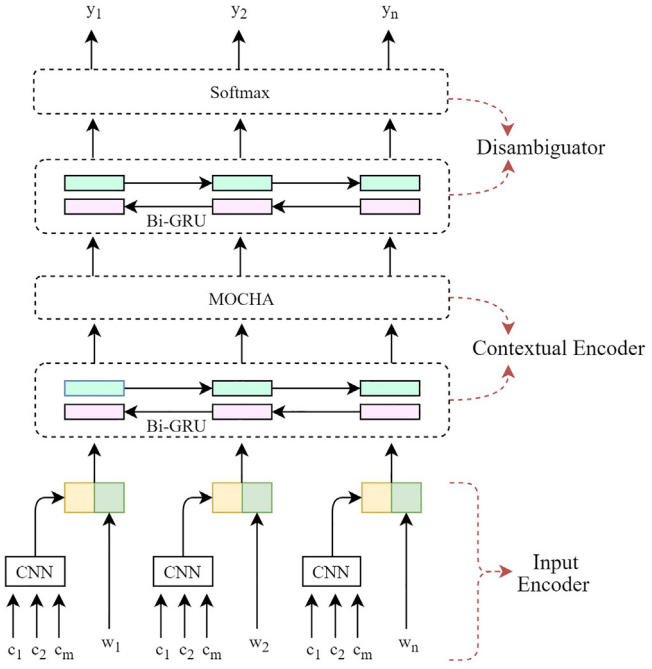

Fig. 2. Overview of the MCCGS architecture with its components.

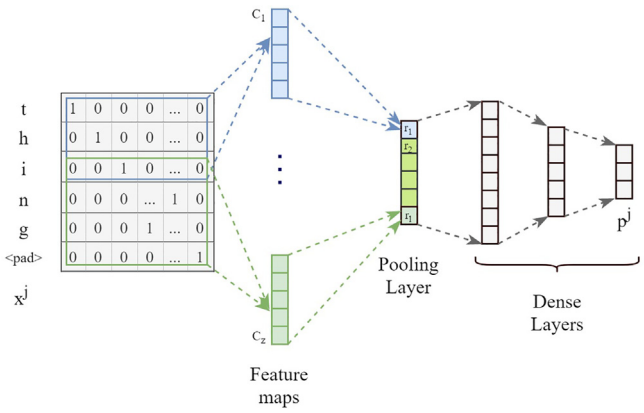

Fig. 3. Character level CNN model architecture to generate word vector.

alignment between input and output, where output is label corresponding to task. The attention is computed over the chunked input sequences.

Let s^{j-1} is disambiguator hidden state at $j-1$, and $h^{1:N} = \left\{h^1, h^2, \ldots, h^N\right\}$ is input hidden vectors. The energy e^{ij} calculated as follows:

$$e^{j,i} = MonotonicEnergy\left(s^{j-1}, h^i\right) \tag{10}$$

Where,

$$MonotonicEnergy\left(s^{j-1}, h^i\right) = g \cdot \frac{v^T}{||v||} \tanh\left(W_s s^{j-1} + W_h h^i + b\right) + r \tag{11}$$

W_s, W_h, r, b, v and g are trainable parameters. The energy scalers for timestamp t_j of the output is obtained from $i = t_{j-1}, t_{j-1}+1, t_{j-1}+2, \ldots, N$. It is passed to the logistic sigmoid function to produces the selection probabilities $p^{j,i}$.

$$p^{j,i} = \sigma(e^{j,i} + \xi), \quad \xi \sim N(0,1) \tag{12}$$

Here, logistic sigmoid with the unit-variance Gaussian noise ξ, constraints selection probabilities $p^{j,i}$ in binary value. The context c^j is generated by the hidden states h^i, which are selected on the $p^{j,i}$. The fixed window length w to hidden states is referred as chunk at here. Chunk energy is calculated in the same way as monotonic energy but skipping the length normalization of v, g and r. The soft attention over preceding w on hidden states and t_j have applied for c^j.

$$v = t_j - w + 1 \tag{13}$$

$$u^{j,k} = ChunkEnergy\left(s^{j-1}, h^k\right); \quad k \in \{v, v+1, v+2, \ldots, t_j\} \tag{14}$$

$$c^j = \sum_{k=v}^{t_j} softmax(u^{j,k}) \cdot h_k \tag{15}$$

3.3. Disambiguator

Most of the earlier work on sequential labeling depends on long short-term memory and conditional random fields to disambiguate the structured inferences (Murthy et al., 2018). The hidden state of each timestamp with the associated generated context vector is used to disambiguate the label dependencies using the bi-directional GRU.

$$\overrightarrow{h^j} = \overrightarrow{GRU}(s^{j-1}, c^j) \tag{16}$$

$$\overleftarrow{h^j} = \overleftarrow{GRU}(s^{j-1}, c^j) \tag{17}$$

$$h^j = \left[\overrightarrow{h^j}; \overleftarrow{h^j}\right] \tag{18}$$

The h^j is passed to a multi layer neural network before to a softmax layer. This penultimate layer predicts the label sequence after scaling and normalizing the output of multi layer neural network.

$$o^j = h^j.W_h + b_h \tag{19}$$

$$P(\hat{y}^i|x;\theta) = \frac{\exp(o^j)}{\sum_{i=1}^{l} \exp(o^j)} \tag{20}$$

Here, W_h and b_h are the learning parameters. The training objective is to minimize the cross-entropy loss (J) along with the l_2 normalization.

$$J_\theta = -\frac{1}{m}\sum_{d=1}^{m}\sum_{l=1}^{q} y^l \log(\hat{y}^l) + (1-y^l)\log(1-\hat{y}^l) + \frac{\lambda}{2}||\theta||^2 \tag{21}$$

For each sentence d, the model predicts q POS labels in a real-valued vector which converted into a one-hot vector used to calculate the average difference with the valid POS labels. It is accommodated by cross-entropy since the probability of each label depends on the probability of another label.

4. Inclusion of morphological features

Deep learning models automatically generate their features according to the given training data and not depend on handcrafted features as were the traditional statistical models. Out of handcrafted features, symbolic features are very common such as lexicon, affixes. Recent studies have empirically proved that the use of these handcrafted external features improves the model performance (Agić et al., 2018; Plank and Klerke, 2019; Scherrer and Rabus, 2019; Gupta et al., 2020). Plank and Klerke (2019) have shown the improvement for POS tagging by incorporating the lexicon features at input encoder in cross-lingual settings. However, the deep learning model supports layered architecture, and each layer captures its features. Hence, we have incorporated morphological features at all three components, i.e. at Input Encoder, Contextual Encoder and Disambiguator of the proposed MCCGS model.

Let $\mathscr{F}$ feature set is available with the training data, which includes $f_1, f_2, f_3, \ldots, f_k$ morphological features. Each feature in a feature set is represented by a vector $(\overrightarrow{f_i})$ to concatenate with each other.

$$\mathscr{F} = \overrightarrow{f_1} \oplus \overrightarrow{f_2} \oplus \overrightarrow{f_3} \oplus \ldots \overrightarrow{f_k} \tag{22}$$

The inclusion of morphological features into the MCCGS model describes in the following way. Here, we have described only changes by which MCCGS model architecture extends after performing the feature inclusion since the remaining components of the MCCGS model are the same.

1. The inclusion of this feature set at input encoder of the MCCGS model is expressed by -

$$w^{1:N} = [p^{1:N}; v^{1:N}] \oplus \mathscr{F}^{1:N} \tag{23}$$

Here, $w^{1:N}$ is the resultant word vector which substituted in the Eq. 6 of the MCCGS model.

2. While integrating this morphological feature set at the contextual encoder, the final output representation through Bi-GRU (mentioned in Eq. 9) has changed with the new representation by -

$$h^{1:N} = \left[\overrightarrow{h}^{1:N}; \overleftarrow{h}^{1:N}\right] \oplus \mathscr{F}^{1:N} \tag{24}$$

3. While including the morphological features at the disambiguator component then input for the multi layer neural network has changed. The new input for this layer (Eq. 18) is represented by -

$$h^j = \left[\overrightarrow{h^j}; \overleftarrow{h^j}\right] \oplus \mathscr{F}^j \tag{25}$$

Here, the feature set of jth word, $\mathscr{F}^j$ is concatenated with the final hidden representation of the Bi-GRU of the disambiguator layer.

Fig. 4 depicts the model architecture encompassing the inclusion of morphological features into the Input-Encoder, Contextual Encoder, and Disambiguator components. The inclusion of morphological features is mutually exclusive; therefore, inclusion is performed at a single component. The schema is an extension of Fig. 2 concerns a feature set $\mathscr{F}$ that consists of the feature vector $\overrightarrow{f_1}, \overrightarrow{f_2}, \overrightarrow{f_3} \ldots \overrightarrow{f_k}$. The feature vectors are concatenated before integrating into each of the components. The output of each concatenator, pink, yellow, and blue, represents Input-Encoder, Contextual Encoder, and Disambiguator components, respectively.

5. Experiments

5.1. Dataset

We have conducted experiments on 21 languages and annotated datasets of these languages obtained from the Universal Dependencies (UD) treebank[1] (version 2.7) to validate the proposed model architecture. The pre-defined splitting of the datasets in the training and development is adopted to train and test our model. The statistics of the training and development dataset is mentioned in Table 2. The entropy of the training and development data is mentioned in Fig. 5, which was calculated at the unigram level by Shannon's entropy (Shannon, 1951; Song et al., 2012; Mundotiya et al., 2020) that shows the nearly symmetrical distribution.

In our experiments, a sentence with their universal part of speech (UPOS) tags and additional features are considered. These additional features are the Tense, Case, Gender, Number, Person and Lemma, of the treebank. These additional features considered as input to make an accurate prediction.

5.2. Experimental settings

The proposed model architecture has three core components which are Input Encoder, Contextual Encoder and Disambiguator. The Input Encoder holds a token vector at a word and character-level information in a latent vector of 100 and 16 dimensions. The word latent vector is randomly initialized through the uniform distribution of $[-0.05, 0.05]$ (Kim et al., 2016), whereas the character level vector initialized with a one-hot vector of dimension 30. The CNN has applied with two convolution layers of size 64, 124 and a fixed window size of 3 as kernel followed by maximum pooling. The number of units in the multilayer feed-forward layer is equal to the size of the convolution layer, which applied over the one-hot vector to obtain the token vector at character-level information (Santos et al., 2014). The Bi-GRU has 128 hidden vectors that capture contextual sentence information. The monotonic chunk-wise attention has carried dependencies among the adjacent words by the chunk size of 10 and emission probability of 0.6. The label side dependencies are captured by Bi-GRU with 128 hidden units in the disambiguator component. We have used an advanced version of the gradient descent learning algorithm with backpropagation through time (BPTT) to optimize the weight for training the model. The goal of BPTT is to update the weights of a neural network to minimize the error compared to some expected output. It is a supervised learning algorithm that allows the network to be corrected concerning already given labels. The Adam optimizer (advanced version of the gradient descent) has been employed to train the model with a 0.01 initial learning rate and 0.007 decay rate. Here, update in learning rate is defined with $\eta_t = \frac{\eta_0}{1+\rho_t}$, where t is referred to the number of completed epochs, η_0 and ρ are the initial learning rate and decay rate, respectively. The batch size and epochs are fixed during the training, i.e., 32 and 40, respectively. The early stoppage (Caruana et al., 2001) with the patience value of 3 is applied to the validation performance to avoid the model overfitting. An additional regularizer, dropout with a value of 0.5 probability, has also been used.

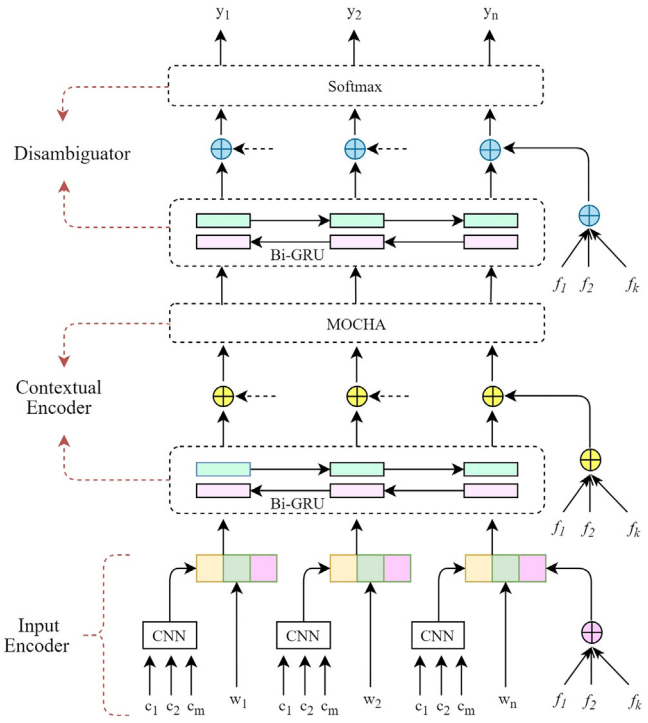

Fig. 4. MCCGS model architecture with symbolic feature inclusion.

Table 2
The languages with their statistics, obtained from the UD treebank. Some languages have more than one treebank; hence the related treebank information has mentioned after - in the language name.

Language	Train Size	Dev Size
Hungarian (hu)	910	441
Greek (el)	1662	403
Swedish-LinES (sv)	3176	1032
Danish (da)	4383	564
Hebrew (he)	5421	484
Croatian (hr)	6914	960
Bulgarian (bg)	8907	1115
Portuguese-GSD (pt)	9664	1210
Dutch-Alpino (nl)	12264	718
English-EWT (en)	12543	2002
Italian-IDST (it)	13121	564
Hindi-HDTB (hi)	13304	1659
German-GSD (de)	13814	799
Spanish-GSD (es)	14187	1400
French-GSD (fr)	14449	1476
Finnish-FTB (fi)	14981	1875
Norwegian-Bokmal (no)	15696	2409
Polish-PDB (pl)	17722	2215
Romanian-Nonstandard (ro)	24122	1052
Persian-PerDT (fa)	26196	1456
Czech-PDT (cs)	68495	9270

[1] https://universaldependencies.org/

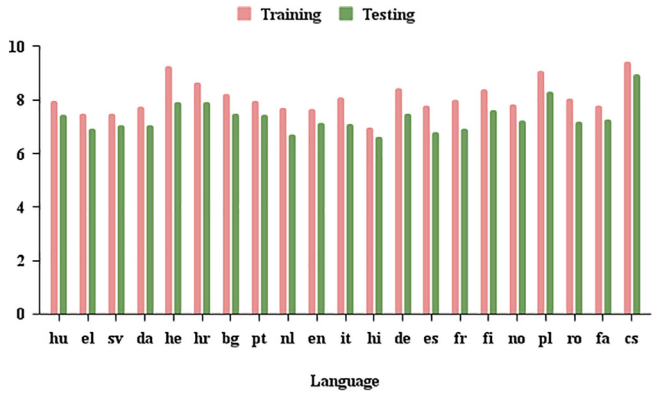

Fig. 5. Entropy of the training and testing datasets.

6. Results and analysis

To compare the performance of our proposed model, Accuracy, Precision, Recall, F1-score and Matthews correlation coefficient (MCC) metrics have been used.

Table 3 presents tagging accuracy for the 21 individual languages on the development data after applying the MCCGS model. In this table, columns represent the Precision, Recall, F1, MCC and Accuracy of each of the corresponding 21 languages. To compare the reported results, we have considered state-of-the-art techniques that have improved their performance without additional features incorporation, Position-aware Self Attention (PSA) mechanism (Wei et al., 2021) and with incorporating additional features that are Type constraints (Täckström et al., 2013), Retrofitting (Faruqui et al., 2015) and Distant Supervision from Disparate Sources (Plank and Agić, 2018). Inclusion of the lexical information by exploiting Type Constraints (TC_w) and Retrofitting (Retro) off-the-shelf embeddings evaluates in the neural tagging literature. Distant Supervision from Disparate Sources (DsDs) (Plank and Agić, 2018) is the alternative way of using lexical information. The replication of features inclusion results presents in the three columns towards the right side of Table 3. For comparison, we are reusing the tagging accuracy values from (Plank and Klerke, 2019). However, the PSA results have been obtained from the entire model training from scratch. In 7 out of 21 languages, the MCCGS model performs better accuracy. The MCCGS model is the best-performing model than retrofitting model in all 21 languages. Our model also gives better accuracy than the TC_w other than two languages (pt and it). The DsDs shows better accuracy than all three models (MCCGS, TC_w, and Retro) to five languages, pt, es, it, bg and de, which are surprising. Whereas, the PSA model reports the highest accuracy on twelve languages.

6.1. Feature inclusion

UD treebank has richness in the lexical and grammatical features of the words since it provides the facility to add language-specific features. Here, we have used those prevalent features which are available in the experimental dataset for all languages, such as Gender, Number, Person, Case, Lemma and Tense. All those features have concatenated before inclusion at the different components of the MCCGS model. The remaining training settings are the same, i.e. mentioned in the parameter settings for the MCCGS model. We can analyze that feature inclusion at disambiguator (MCCGS-D) improves the accuracy compared to the rest of the MCCGS feature inclusion for most of the languages from Table 4. en is the only language that decreases the model performance by 1% after utilizing such features. Like this, cs is the only language on which feature inclusion at contextual encoder (MCCGS-CE) improves the accuracy by 0.59%. Feature inclusion at input encoder (MCCGS-IE) improves the accuracy of nl, pt, es, da, fr and de languages by $1.87\%, 4.64\%, 1.78\%, 0.92\%, 1.52\%$ and 3.14%, respectively. There are four languages (es, it, bg and de) on which the DsDs model performs better compared to the MCCGS model, as interpreted by Table 3. Although, MCCGS-IE further improves accuracy for bg and de languages mentioned in Table 4.

The proposed model, MCCGS and its variants have been compared with the TC_w, Retro, DsDs and PSA model to show the significance of the results by single factor ANalysis Of VAriance (ANOVA) test. The MCCGS, MCCGS-IE, MCCGS-CE and MCCGS-D have obtained the p-values, $0.003, 0.001, 0.002$ and 0.000, respectively. These p-values are comparatively low than the significance value (0.05), which emits stronger evidence against the earlier hypothesis made by state-of-the-art models. This conclusion supports our obtained results, mentioned in Table 3 and Table 4.

The mean accuracy of the model compared to their variants and some earlier state-of-the-art models show that MCCGS-D provides the highest score, 92.3%. Whereas the proposed MCCGS model provides mean accuracy 90.4%, further improving by feature inclusion. The state-of-the-art models provide $83.6\%, 81.3\%, 84.1\%$ and

Table 3
Obtained results using the MCCGS model.

Lang	Precision	Recall	F1	MCC	Accuracy	TC_w	Retro	DsDs	PSA
hu	83.8	79.59	80.02	75.97	79.59	77.5	75.5	76.2	86.20
nl	93.68	91.44	91.64	91.95	91.44	89.2	86.6	89.6	91.85
pt	91.35	89.95	90.20	89.53	89.95	92.2	88.6	93.1	91.97
es	90.73	89.66	88.99	89.02	89.66	88.9	88.9	91.7	64.58
hi	94.61	94.57	94.47	94.11	94.57	63.9	63.0	66.2	94.85
it	88.57	89.44	88.73	88.26	89.44	91.8	90.0	93.7	88.88
da	91.00	90.55	90.38	90.08	90.55	89.3	88.2	90.1	89.08
fi	93.89	93.13	93.14	92.86	93.13	81.4	79.2	83.1	92.43
el	86.68	87.00	85.77	85.47	87.00	86.1	79.3	79.2	89.55
bg	89.20	90.04	88.93	89.42	90.04	89.9	87.1	91.0	92.46
cs	97.23	95.53	95.75	95.84	95.53	87.5	84.9	87.4	95.14
he	85.12	82.53	82.75	84.67	82.53	75.9	71.7	76.8	88.74
no	96.12	95.55	95.40	93.88	95.55	91.1	88.8	91.4	94.66
fa	95.53	94.90	95.11	94.88	94.90	43.8	44.1	43.6	95.87
sv	92.53	92.22	92.06	92.24	92.22	89.2	87.0	89.8	93.09
en	94.58	94.74	94.36	88.51	94.74	87.6	82.5	87.3	94.02
ro	88.74	86.07	85.47	85.71	86.07	84.2	80.2	86.0	88.02
hr	91.33	89.89	89.60	88.98	89.89	85.2	83.0	85.9	90.50
fr	93.75	93.20	92.52	92.61	93.20	90.0	89.9	91.3	89.39
de	88.41	87.20	86.97	85.43	87.20	87.1	84.7	87.5	90.20
pl	93.87	92.77	92.01	92.29	92.77	84.9	83.9	85.4	92.35

Table 4
Obtained scores after inclusion of morphological features in the MCCGS model. Here, IE, CE and D stand for MCCGS-IE, MCCGS-CE and MCCGS-D models respectively.

	Precision			Recall			F-score			MCC			Accuracy		
Lang	**IE**	**CE**	**D**	**IE**	**CE**	**D**	**IE**	**CE**	**D**	**IE**	**CE**	**D**	**IE**	**CE**	**D**
hu	81.07	82.89	82.79	81.33	80.00	82.61	79.93	79.95	81.74	77.97	75.84	82.95	81.33	80.00	82.61
nl	92.99	92.37	93.87	93.31	92.18	92.84	92.97	91.63	92.71	91.20	89.48	89.95	93.31	92.18	92.84
pt	94.46	91.25	94.17	94.59	89.81	94.24	94.48	89.99	94.17	94.08	89.74	93.93	94.59	89.81	94.24
es	91.21	91.36	91.12	91.44	90.76	91.39	91.17	90.17	91.08	91.68	91.15	90.71	91.44	90.76	91.39
hi	93.25	94.21	94.81	93.09	93.91	94.77	92.99	93.92	94.63	92.65	93.60	94.42	93.09	93.91	94.77
it	90.23	90.69	92.01	90.91	90.73	92.21	90.33	90.28	91.67	89.88	90.01	91.56	90.91	90.73	92.21
da	91.10	88.76	91.48	91.47	89.24	91.24	91.10	88.60	90.53	91.08	88.29	90.59	91.47	89.24	91.24
fi	93.10	90.78	93.54	93.01	90.92	93.75	92.43	90.51	93.00	92.24	90.11	93.14	93.01	90.92	93.75
el	86.95	90.86	89.63	87.48	87.75	89.53	86.80	87.84	88.34	86.28	87.90	88.35	87.48	87.75	89.53
bg	89.63	86.67	91.85	89.40	88.25	92.55	88.72	86.77	91.72	91.50	85.04	91.70	89.40	88.25	92.55
cs	96.31	96.79	95.76	95.73	96.12	95.80	95.87	96.21	95.57	95.76	96.14	95.49	95.73	96.12	95.80
he	84.38	87.62	87.80	84.68	87.77	87.91	83.65	87.01	87.21	84.84	79.63	86.77	84.68	87.77	87.91
no	96.34	96.54	96.45	95.57	96.25	96.28	95.69	96.30	96.09	95.65	95.95	95.76	95.57	96.25	96.28
fa	95.83	95.83	95.79	95.63	95.73	95.99	95.59	95.60	95.71	95.42	95.42	95.58	95.63	95.73	95.99
sv	93.03	93.79	94.74	92.82	91.04	94.76	92.73	91.44	94.50	92.12	91.22	94.32	92.82	91.04	94.76
en	93.33	93.35	93.95	93.50	93.31	93.74	93.28	93.06	93.71	89.70	87.86	91.82	93.50	93.31	93.74
ro	87.41	88.29	90.39	87.85	86.63	91.16	87.28	85.98	90.31	88.86	88.60	90.10	87.85	86.63	91.16
hr	89.86	91.50	90.55	89.30	89.75	90.41	89.24	89.71	90.00	86.18	89.36	89.72	89.3	89.75	90.41
fr	94.74	94.97	92.77	94.72	94.33	92.97	94.53	94.44	92.53	94.33	94.24	92.35	94.72	94.33	92.97
de	91.01	91.85	91.45	90.43	90.34	90.07	90.70	90.09	88.51	89.34	90.08	89.07	90.43	90.34	90.07
pl	94.15	94.31	95.28	94.14	93.90	95.34	93.90	93.50	94.95	93.75	93.48	94.94	94.14	93.90	95.34

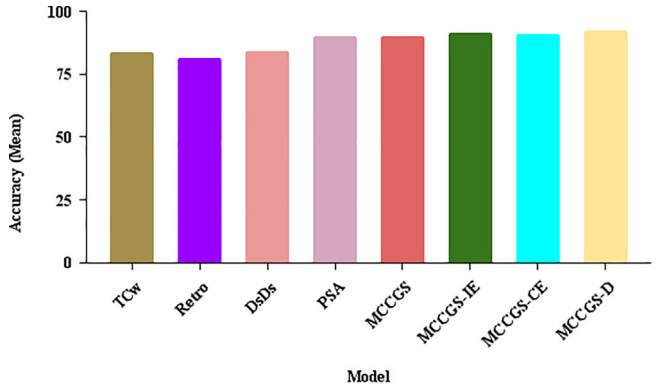

Fig. 6. Comparison of mean accuracy scores.

90.18% mean accuracy by TC_w, Retro, DsDs and PSA, respectively. The comparison of the mean accuracy among those models is mentioned in Fig. 6.

6.2. Effect of chunk size

One of the hyper-parameters of the MCCGS model has chunk size *w*, mainly used to capture the word information from adjacent words. It allows managing non-continuous relations among the words of a selected chunk. For this claim, we have performed extensive experiments with different chunk sizes for each model. Table 5 shows the acquired mean accuracy results of the MCCGS with its variants, MCCGS-IE, MCCGS-CE, MCCGS-D, as the chunk size of 1 to 10. It is empirical proof that chunk size increases will increase the mean accuracy for all the models. This incremental rate has fallen after the window size 9 due to obtaining consistent accuracy.

6.3. Analysis

Here we present an analysis of the proposed MCCGS and its variations to understand the influence of feature inclusion better. For this realization, a sample text is taken from the development set of English-EWT treebank for simulating performance, as mentioned in Fig. 7. Expressing past information of the sentence, the reinstating conjunction 'when' is used to connect two clauses. The subordinate clause 'came' indicates the features as tense is past, mood indefinite, and sentence type is finite. The same features signify the matrix clause 'arrested'. Even though these two are different clauses connected by the conjunction can realize long-distance relations.

Greenberg's language universal (Greenberg et al., 1963), a noun (NN) is always followed by an adjective (JJ). According to these hypotheses, the training data followed a combination of noun phrases (JJ + NN). As a result, the model predicts JJ + NN in most of the cases instead of JJ + JJ. Fig. 7 example also shows the same combination, which led to the wrong prediction from the proposed models. Same with the verb phrase as well, the verb phrase contains RB + VBN while nouns directly preceded by the verb in this scenario from the proposed model. The impact of additional features can be seen in the model prediction. The MCCGS model predicts wrong output for adjacent words. For example, 'briefly arrested' should be an adverb (RB) and verb (VDB), but the model predicts adjective (JJ) and noun (NN), respectively. After using the features during the model training, we found that the number of errors has drastically improved. Here, the 'arrested' is getting tense information (Tense = Past) from the 'came' and it carries forward in the other morphologically similar words. Due to this, the MCCGS-IE, MCCGS-CE and MCCGS-D predict RB + NN, VBD + VBD and RB + VBD, respectively, for the 'briefly arrested'. The feature inclusion near the label prediction provides an accurate result for capturing longer dependencies among words.

6.4. Experiments on indo-aryan languages

To show the effectiveness of our proposed model to low resource languages, the languages which belong to Indo-Aryan languages in the UD treebank have been exploited for the experiments. In this, Marathi (mr) (UFAL treebank), Tamil (ta) (TTB treebank), Telugu (te) (MTG treebank) and Sanskrit (sa) (Vedic treebank) languages with their splits, i.e. 373, 400, 1051, and 2524 sentences to respective languages for the training set and 46, 80, 131 and 1473 sentences to respective languages for the development set have evaluated. Similarly, another treebank based on Paninian Grammar Framework, Hindi-Urdu multi-

Table 5
Effect of the chunk size on the model performance in terms of mean accuracy.

Chunk Size	1	2	3	4	5	6	7	8	9	10
MCCGS	86.13	86.78	87.51	88.24	88.67	89.12	89.94	90.00	90.19	90.40
MCCGS-IE	86.94	87.83	88.67	89.15	89.85	89.65	89.98	90.51	91.39	91.40
MCCGS-CE	85.03	86.14	87.09	88.40	88.93	89.95	89.47	90.40	90.61	90.90
MCCGS-D	87.67	87.45	88.10	88.95	89.31	90.14	90.87	91.32	91.98	92.30

Sentence	In	Fallujah	,	hundreds	of	demonstrators	came	out	against	US	troops	when	they	briefly	arrested	a	young	newlywed	bride	.
Actual Label	IN	NNP	,	NNS	IN	NNS	VBD	RB	IN	NNP	NN	WRB	PRP	RB	VBD	DT	JJ	JJ	NN	.
MCCGS	IN	NNP	,	NNS	IN	RB	VBD	IN	IN	NNP	NN	WRB	PRP	JJ	NN	DT	JJ	JJ	NN	.
MCCGS-IE	IN	NNP	,	NNS	IN	NNS	VBD	IN	IN	NNP	NN	WRB	PRP	RB	NN	DT	JJ	JJ	NN	.
MCCGS-CE	IN	NNP	,	NNS	IN	RB	VBD	IN	IN	NNP	NN	WRB	PRP	VBD	VBD	DT	JJ	JJ	NN	.
MCCGS-D	IN	NNP	,	NNS	IN	NNS	VBD	RB	IN	NNP	NN	WRB	PRP	RB	VBD	DT	JJ	JJ	NN	.

Fig. 7. Example of models prediction to POS tagging.

Table 6
Obtained scores on the Indo-Aryan languages of UD treebank and HUTB dataset.

		UD treebank (Lang)				HUTB (Lang)	
	Models	**mr**	**ta**	**te**	**sa**	**hi**	**ur**
Precision	MCCGS	80.20	78.56	91.60	96.64	95.68	86.93
	MCCGS-IE	88.16	79.73	94.68	97.91	94.10	84.74
	MCCGS-CE	81.44	79.62	94.12	97.03	94.85	87.49
	MCCGS-D	93.33	87.25	96.02	97.29	94.45	86.95
Recall	MCCGS	85.03	79.80	93.29	95.79	95.60	87.06
	MCCGS-IE	89.09	82.17	95.20	96.23	94.78	85.38
	MCCGS-CE	85.22	80.15	94.20	95.46	94.59	87.62
	MCCGS-D	93.58	87.50	96.69	97.28	94.52	87.48
F1	MCCGS	82.37	77.79	92.38	95.66	95.60	87.06
	MCCGS-IE	87.33	80.18	94.67	96.56	94.78	85.38
	MCCGS-CE	81.50	78.70	93.42	95.54	94.59	87.62
	MCCGS-D	93.09	86.90	96.25	97.27	94.52	87.48
MCC	MCCGS	77.32	76.46	76.58	92.83	94.16	86.67
	MCCGS-IE	73.71	82.02	90.73	94.68	94.23	86.90
	MCCGS-CE	82.34	74.42	92.07	95.75	94.10	88.63
	MCCGS-D	92.99	87.04	96.23	96.73	93.93	86.13
Accuracy	MCCGS	85.03	79.80	93.29	95.79	95.64	86.81
	MCCGS-IE	89.09	79.73	95.20	96.23	94.41	84.24
	MCCGS-CE	85.22	80.15	94.20	95.46	94.57	87.47
	MCCGS-D	93.58	87.50	96.69	97.28	94.25	86.76

representational treebank[2] (HUTB), has been evaluated. From this treebank, the Hindi and Urdu model trained on 16997, 5648 sentences and tested on 1910, 635 sentences, respectively (Bhatt et al., 2009; Bhat et al., 2017). The Hindi treebank dataset is belonging to News and Articles domain.

The MCCGS-D model performs better compared to the rest of the model for all these Indo-Aryan languages, as shown in Table 6. However, significant improvements in accuracy have been shown for the mr and ta languages which are 8.55 and 7.7 for the respective languages, even these comprising minimalist training data. Although this significant improvement also exists for the Precision, Recall, F1 and MCC score as well. For the sa language, this improvement is minimum, 1.49 compared to mr, ta, te. Similarly, the MCCGS-CE model performs adequately on ur language of the HUTB dataset compared to other models. Since hi languages of the HUTB dataset have a sufficient number of an annotated sentences, the MCCGS model performs better.

7. Conclusions

To address the problem of contextual importance in the deep learning-based model to part of speech tagging, we have proposed a solution with monotonic chunk attention, namely Monotonic Chunk-wise Attention with CNN-GRU-SOFTMAX (MCCGS). According to their functionality, the proposed model architecture has three essential components according to their functionality, i.e. character and word-level information extraction done by Information Encoder, capturing sequential information among words with higher importance to context words captured by the Contextual Encoder and label side disambiguation done by Disambiguator. Additionally, the inclusion of morphological features (Gender, Number, Person, Case, Lemma and Tense) has also been performed

[2] http://ltrc.iiit.ac.in/hutb_release/

on different components to determine its importance and features. These features are pervasive and exist in the Universal Dependencies treebank. Extensive experiments exhibit the advantage of MCCGS over existing models on the dataset of 21 languages, available on Universal Dependency treebank with feature integration. The MCCGS model improves the mean accuracy score compared to the existing state-of-the-art models. The highest gain of feature inclusion has observed at the unification of features at the Disambiguator component (MCCGS-D) that is 1.9% compared to the MCCGS model. The same MCCGS-D model has reported the highest score for the lower annotated dataset of the Indo-Aryan languages. The proposed models have validated with the Hindi and Urdu languages belonging to another Hindi-Urdu multi-representational treebank (HUTB) dataset. This attention based model architecture extracts the non-continuous relations among the words of the selected chunk, which is a notable constraint of this model for highly morphological agglutinative languages. The proposed model architecture can also improve the performance of various language-based systems such as second-language learning, machine translation and text-to-speech synthesis for low resource languages.

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

Agić, Ž., Plank, B., 2018. Distant supervision from disparate sources for low-resource part-of-speech tagging, in. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP), p. 614.

Altun, Y., Tsochantaridis, I., Hofmann, T., 2003. Hidden markov support vector machines, in: Proceedings of the 20th international conference on machine learning (ICML-03), pp. 3–10..

Bahdanau, D., Cho, K., Bengio, Y., 2015. Neural machine translation by jointly learning to align and translate, in: Bengio, Y., LeCun, Y. (Eds.), 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015, Conference Track Proceedings..

Bhat, R.A., Bhatt, R., Farudi, A., Klassen, P., Narasimhan, B., Palmer, M., Rambow, O., Sharma, D.M., Vaidya, A., Vishnu, S.R., et al., 2017. The hindi/urdu treebank project. In: Handbook of Linguistic Annotation. Springer, pp. 659–697.

Bhatt, R., Narasimhan, B., Palmer, M., Rambow, O., Sharma, D.M., Xia, F., 2009. A multi-representational and multi-layered treebank for hindi/urdu, in. In: Proceedings of the Third Linguistic Annotation Workshop (LAW III), pp. 186–189.

Caruana, R., Lawrence, S., Giles, L., 2001. Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping. Adv. Neural Inform. Processing Systems, 402–408.

Chakrabarty, A., Chaturvedi, A., Garain, U., 2019. Neumorph: Neural morphological tagging for low-resource languages–an experimental study for indic languages. ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP) 19, 1–19.

Chiu, C., Raffel, C., 2018. Monotonic chunkwise attention, in: 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings, OpenReview.net. URL: https://openreview.net/forum?id=Hko85plCW..

Devlin, J., Chang, M.W., Lee, K., Toutanova, K., 2019. BERT: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota. pp. 4171–4186. URL:https://www.aclweb.org/anthology/N19-1423, doi: 10.18653/v1/N19-1423..

Dos Santos, C., Zadrozny, B., 2014. Learning character-level representations for part-of-speech tagging. International Conference on Machine Learning, PMLR., 1818–1826

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..

Faruqui, M., Dodge, J., Jauhar, S.K., Dyer, C., Hovy, E., Smith, N.A., 2015. Retrofitting word vectors to semantic lexicons, in: Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, Denver, Colorado. pp. 1606–1615. URL:https://www.aclweb.org/anthology/N15-1184, doi: 10.3115/v1/N15-1184..

Greenberg, J.H. et al., 1963. Some universals of grammar with particular reference to the order of meaningful elements. Universals of language 2, 73–113.

Gupta, A., Krishna, A., Goyal, P., Hellwig, O., 2020. Evaluating neural morphological taggers for Sanskrit, in: Proceedings of the 17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology, Association for Computational Linguistics, Online. pp. 198–203. URL:https://www.aclweb.org/anthology/2020.sigmorphon-1.23, doi: 10.18653/v1/2020.sigmorphon-1.23..

Hinton, G.E., Osindero, S., Teh, Y.W., 2006. A fast learning algorithm for deep belief nets. Neural computation 18, 1527–1554.

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

Indurthi, S.R., Chung, I., Kim, S., 2019. Look harder: A neural machine translation model with hard attention, in. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 3037–3043.

Kim, Y., Jernite, Y., Sontag, D., Rush, A., 2016. Character-aware neural language models. In: in: Proceedings of the AAAI conference on artificial intelligence.

Kudo, T., Matsumoto, Y., 2001. Chunking with support vector machines, in: Second Meeting of the North American Chapter of the Association for Computational Linguistics..

Kupiec, J., 1992. Robust part-of-speech tagging using a hidden markov model. Computer speech & language 6, 225–242.

Lafferty, J.D., McCallum, A., Pereira, F.C.N., 2001. In: Conditional random fields: Probabilistic models for segmenting and labeling sequence data, in Proceedings of the Eighteenth International Conference on Machine Learning. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 282–289.

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..

Lin, J.C.W., Shao, Y., Djenouri, Y., Yun, U., 2021. Asrnn: a recurrent neural network with an attention model for sequence labeling. Knowledge-Based Systems 212, 106548.

Ling, W., Dyer, C., Black, A.W., Trancoso, I., Fermandez, R., Amir, S., Marujo, L., Luís, T., 2015. Finding function in form: Compositional character models for open vocabulary word representation. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Lisbon, Portugal, pp. 1520–1530. https://doi.org/10.18653/v1/D15-1176. URL:https://www.aclweb.org/anthology/D15-1176.

Liu, L., Shang, J., Ren, X., Xu, F., Gui, H., Peng, J., Han, J., 2018. Empower sequence labeling with task-aware neural language model. In: Proceedings of the AAAI Conference on Artificial Intelligence.

Lu, B.L., Ma, Q., Ichikawa, M., Isahara, H., 2003. Efficient part-of-speech tagging with a min-max modular neural-network model. Applied Intelligence 19, 65–81.

Ma, X., Hovy, E., 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 (Volume 1: Long Papers), Association for Computational Linguistics, Berlin, Germany. pp. 1064–1074. URL:https://www.aclweb.org/anthology/P16-1101, doi: 10.18653/v1/P16-1101..

Mundotiya, R.K., Kumar, V., Mehta, A., Singh, A.K., 2020. Attention-based domain adaption using transfer learning for part-of-speech tagging: An experiment on the hindi language. In: Proceedings of the 34th Pacific Asia Conference on Language. Information and Computation, pp. 471–477.

Mundotiya, R.K., Singh, M.K., Kapur, R., Mishra, S., Singh, A.K., 2020b. Basic linguistic resources and baselines for bhojpuri, magahi and maithili for natural language processing. CoRR abs/2004.13945. URL:https://arxiv.org/abs/2004.13945, arXiv:2004.13945..

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, 1–20.

Niehues, J., Cho, E., 2017. Exploiting linguistic resources for neural machine translation using multi-task learning, in. In: Proceedings of the Second Conference on Machine Translation, pp. 80–89.

Plank, B., Agić, Ž., 2018. Distant supervision from disparate sources for low-resource part-of-speech tagging. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Brussels, Belgium, pp. 614–620. https://doi.org/10.18653/v1/D18-1061. URL:https://www.aclweb.org/anthology/D18-1061.

Plank, B., Klerke, S., 2019. Lexical resources for low-resource pos tagging in neural times, in. In: Proceedings of the 22nd Nordic Conference on Computational Linguistics, pp. 25–34.

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.

Ratnaparkhi, A., 1996. A maximum entropy model for part-of-speech tagging, in: Conference on empirical methods in natural language processing..

Sagot, B., Alonso, H.M., 2017. Improving neural tagging with lexical information, in. In: Proceedings of the 15th International Conference on Parsing Technologies, pp. 25–31.

dos Santos, C.N., Zadrozny, B., 2014. Learning character-level representations for part-of-speech tagging, in. 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.

Sarawagi, S., Cohen, W.W., 2004. Semi-markov conditional random fields for information extraction. Adv. Neural Information Processing Systems 17, 1185–1192.

Scherrer, Y., Rabus, A., 2019. Neural morphosyntactic tagging for rusyn. Natural Language Eng. 25, 633–650.

Shannon, C.E., 1951. Prediction and entropy of printed english. Bell System Tech. J. 30, 50–64.

Shao, Y., Lin, J.C.W., Srivastava, G., Jolfaei, A., Guo, D., Hu, Y., 2021. Self-attention-based conditional random fields latent variables model for sequence labeling. Pattern Recognition Letters 145, 157–164.

Sinha, K., Dong, Y., Cheung, J.C.K., Ruths, D., 2018. A hierarchical neural attention-based text classifier. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Brussels, Belgium, pp. 817–823. https://doi.org/10.18653/v1/D18-1094. URL:https://www.aclweb.org/anthology/D18-1094.

Song, Y., Klassen, P., Xia, F., Kit, C., 2012. Entropy-based training data selection for domain adaptation, in: Kay, M., Boitet, C. (Eds.), COLING 2012, 24th International Conference on Computational Linguistics, Proceedings of the Conference: Posters, 8–15 December 2012, Mumbai, India, Indian Institute of Technology Bombay. pp. 1191–1200. URL:https://www.aclweb.org/anthology/C12-2116/..

Sun, X., Lu, W., 2020. Understanding attention for text classification, in. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 3418–3428.

Täckström, O., Das, D., Petrov, S., McDonald, R., Nivre, J., 2013. Token and type constraints for cross-lingual part-of-speech tagging. Transactions of the Association for. Comput. Linguistics 1, 1–12. URL:https://www.aclweb.org/anthology/Q13-1001.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L.u., Polosukhin, I., 2017. Attention is all you need, in: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (Eds.), Advances in Neural Information Processing Systems, Curran Associates, Inc. URL:https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf..

Wei, W., Wang, Z., Mao, X., Zhou, G., Zhou, P., Jiang, S., 2021. Position-aware self-attention based neural sequence labeling. Pattern Recognition 110, 107636.

Yin, Y., Su, J., Wen, H., Zeng, J., Liu, Y., Chen, Y., 2019. Pos tag-enhanced coarse-to-fine attention for neural machine translation. ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP) 18, 1–14.

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

Zheng, H., Fu, J., Mei, T., Luo, J., 2017. Learning multi-attention convolutional neural network for fine-grained image recognition. In: Proceedings of the IEEE international conference on computer vision, pp. 5209–5217.