J Inf Process Syst, Vol.19, No.3, pp.355~369, June 2023
https://doi.org/10.3745/JIPS.02.0197

ISSN 1976-913X (Print)
ISSN 2092-805X (Electronic)

JOURNAL OF INFORMATION PROCESSING SYSTEMS JIPS

Improved Character-Based Neural Network for POS Tagging on Morphologically Rich Languages

Samat Ali and Alim Murat*

Abstract

Since the widespread adoption of deep-learning and related distributed representation, there have been substantial advancements in part-of-speech (POS) tagging for many languages. When training word representations, morphology and shape are typically ignored, as these representations rely primarily on collecting syntactic and semantic aspects of words. However, for tasks like POS tagging, notably in morphologically rich and resource-limited language environments, the intra-word information is essential. In this study, we introduce a deep neural network (DNN) for POS tagging that learns character-level word representations and combines them with general word representations. Using the proposed approach and omitting hand-crafted features, we achieve 90.47%, 80.16%, and 79.32% accuracy on our own dataset for three morphologically rich languages: Uyghur, Uzbek, and Kyrgyz. The experimental results reveal that the presented character-based strategy greatly improves POS tagging performance for several morphologically rich languages (MRL) where character information is significant. Furthermore, when compared to the previously reported state-of-the-art POS tagging results for Turkish on the METU Turkish Treebank dataset, the proposed approach improved on the prior work slightly. As a result, the experimental results indicate that character-based representations outperform word-level representations for MRL performance. Our technique is also robust towards the-out-of-vocabulary issues and performs better on manually edited text.

Keywords

Character Representation, Deep Neural Network, Morphologically Rich Language, POS Tagging

1. Introduction

Part-of-speech (POS) tagging is a crucial natural language processing (NLP) topic that intends to assign POS tags to each word in a phrase. Generally, a POS is considered as a grammatical classification that includes verbs, adjectives, adverbs, noun, etc. It serves as a preliminary task for performing tasks such as chunking, dependency parsing, named entity recognition on any language. POS tagging on the most used languages, like English and Chinese have seen dramatic improvements with the aid of recent deep learning techniques. Meanwhile, the rapid advancement in the Web and social media have also led to an increase in demand for highly accurate POS tagging for more minority languages that are low-resource and morphologically complex.

Minority languages such as Uyghur, Uzbek and Kyrgyz widely used in Xinjiang and Central Asia and have similar morphology, and they are typical inflectional language that produces various forms of words.

Manuscript received September 1, 2022; first revision October 17, 2022; accepted October 29, 2022.
* **Corresponding Author:** Alim Murat (a.murat@xjnu.edu.cn)
School of Computer Science Technology, Xinjiang Normal University, Urumqi, China (345042453@qq.com, a.murat@xjnu.edu.cn)

In those languages, a word often tends to encode various grammatical properties such as number, person, case, gender, tense, aspect, etc. and can exhibit a large diversity of morphological complexity [1]. However, these complex morpho-syntactic property has generated many challenges to the basic down-stream NLP tasks in those languages. As with many other NLP tasks, POS tagging always severely suffers from unknown or out-of-vocabulary (OOV) word problem and possibly more than others, due to the scarce training data with syntactic annotation. This issue often happens if a language is morphologically so rich and derivative that has various agglutinative property and limited supervised data [2].

Many research studies have suggested that tackling the OOV issue needs to pretrain the word-level embedding, e.g., with word2vec [3], glove [4], and fastText [5]. This has two major advantages: (1) more word tokens, meaning that the vocabulary is expanded by more data, thereby learning representation of more OOV words; (2) more word token per type indicates the better modeling of semantic and syntactic similarities between words. However, it suffers from many limitations:

- Word-level embeddings are ineffective at capturing intra-word information, as they are mainly concerned with word forms.
- Word-level embeddings are unstable for representing the unseen or unnormalized words.

Though, the OOV problem can be resolved by expanding the vocabulary for word embeddings in many commonly used languages, however, it does lack the ability to extract complex morphological information from a word and cannot effectively learn the intra-grammatical structure of the words from the morphologically rich language (MRL). To better generalize the OOV words and to improve the limitations of word-level embeddings for POS tagging performance, we propose to apply the character-based convolutional neural network (CNN) to POS tagging of three MRL. More specifically, our contributions in this work are:

- In view of the Uyghur, Uzbek and Kyrgyz language characteristics, a character-based neural network architecture (CharNN), which adopts a convolutional layer that is highly efficient in extracting morphological features of words of any length is proposed.
- Regarding the lack of publicly available POS tagging dataset in those three languages, the custom datasets (i.e., MultiPOS_ukg) are built and used to verify the effectiveness of our model. Moreover, our model is experimented on the other agglutinative and non-agglutinative language datasets.
- To evaluate the robustness of our model against the OOV problem, the proposed model is performed on the manually unnormalized development-set which is generated by altering the characters of word in the normal development-sets. With the increasing degree of editing operation, the performance of the CharNN degrades much slower than other model, which confirms that the proposed model is more robust against the unnormalized case.

Further experimental results indicate that the proposed CharNN performs well on agglutinative languages, where the OOV word cases highly exist, and it can be adopted by many low-resource NLP systems that are in demand for tackling the OOV words. Moreover, the proposed method can become an additional preprocessing module for obtaining morphological information.

The following section of article is arranged as follows. Section 2 gives a survey on related work. Section 3 describes the method. Section 4 presents the experimental datasets and details the experiments. Section 5 demonstrates the limitation and conclusion.

2. Related Work

Traditional POS tagging pipelines rely on numerous handcrafted features and task-wise knowledge. For agglutinative and resource-limited languages, this type of technique is not very effective. Very recently, deep neural networks (DNNs) have demonstrated great performance in the POS tagging task, owing to their ability to automatically extract syntactic dependencies and semantic information for a word sequence at a high level [6]. Numerous DNN methods of high-efficiency and high performance have been explored for MRL POS tagging in the previous research. Here, some of the common sequential deep learning approaches for POS tagging on MRL are addressed in this section.

Deshmukh et al. [7] have proposed a bidirectional long-short term memory (BiLSTM) model for Marathi POS tagging. In this work, the proposed model can tackle each point from the future and past context of sequence and improved the machine learning techniques by a large margin with an accuracy up to 85% and 97%.

Akhil et al. [8] have proposed a deep learning approach for POS tagging for Malayalam which is Indic language family. The used methods are long-short term memory (LSTM), gated recurrent unit (GRU), and BiLSTM. In the experiment, the dataset was a publicly released tagged dataset for Malayalam, which consists of 287588 words and whose tagset contains 36 tags. The results showed that, out of all architectures, BiLSTM with 64 hidden layers reported the best score with 98.33% F1-measure.

Khan et al. [9] have demonstrated deep learning model for POS tagging task in Urdu. In the experiments, the BJ dataset which has 5,000 sentences with 164,466 words in total was used. The result proved that better performance for Urdu POS was obtained by employing different RNN models with both sparse and dense features, compared to language-independent feature based conditional random field (CRF).

Bahcevan et al. [10] have used a recurrent neural network (RNN) and LSTM model to address the POS tagging for Turkish language. This study used the IMST Universal Dependencies TreeBank which consists of 5,635 sentences with 48,000 words collected from daily news reports and novels. The performance was compared to the state-of-the-art methods. The results showed that LSTM outperforms RNN with 88.7% F1-score.

Todi et al. [11] have presented a statistical POS tagger for Kananda using different machine learning and NN models. The architecture started with the Vanilla RNN network and then used different recurrent architectures. To overcome the OOV words issue, this work switched word-embedding to character embedding and learned a better representation. The results showed deep learning model outperformed the state-of-the-art POS tagger by up to 6%.

Kumar et al. [12] have evaluated the various sequential deep learning method like RNN, GRU, LSTM, and BiLSTM on the manually tagged tweets data for Malayalam POS tagging. The model training phase includes both word and character level representation. It was found that GRU-based model at the word-level produced the highest F1-score of 79.39%, and the BiLSTM performed better at character-level with a F1-score of 82.39%.

Though DNN have successfully been used for POS tagging tasks in recent studies, but most of the research work relatively employed the different variants of RNN-based architectures that needs high-quality word embeddings pretrained on a large amount of corpus, and not many of them discussed the OOV problem. Very little work has been done on Uyghur, Uzbek, and Kyrgyz POS tagging, due to the

scarcity of quality annotated data. One of such endeavor was by Maimaiti et al. [13] where they revealed that the BiLSTM network with a CRF layer achieved accuracy of 98.41% on their dataset only for Uyghur. Moreover, they had used more data and done complex feature engineering to learn the word sequence.

Inspired by the state-of-the-art approach [1], the use of character-based CNN to capture the complex features hidden in the words drew our attention. In their work, the empirical evidence was highlighted that the CNN model shows higher flexibility than LSTM in processing sub-word information.

The CNN architectures with great discriminative ability are generally used in image processing applications [14-16] and have achieved many remarkable performances. In the text processing, the CNN model often utilizes several filters to capture n-grams of any lengths in a word, which can encode any patterns from morphemes up to words in POS tagging. Therefore, this capability of CNN architecture hints that it can work well on agglutinative languages, where the OOV words highly occurs. To this end, we propose a character-based DNN architecture that uses character-based representation to allow more efficient feature extraction from word of any size, hence it eases the problem of Unknow or OOV words in the tagging.

3. Method

The proposed architecture consists of a feature representation model and a scoring model. The feature representation model is a combination of character-level and word-level representation. The scoring model is a network, which scores each word for its tag in a sentence and adopt the Viterbi algorithms to predict specific tags. We explain the details of our method in the following subsections.

3.1 Feature Representation

In the proposed architecture, each word from the input sequence is represented by a combination of two vector features to preserve both semantic and morpho-syntactic granularities of a word. The initial layer of the network generates real-valued feature vectors by word transformation, which can encode pivotal information like morpho-syntactic and semantic properties. We employ a fixed-length word vocabulary V^w and segment words into characters, to build a fixed-length character vocab-table V^{ch}. Considering a sentence that is composed of N words $\{w_1, w_2, \dots, w_N\}$, each word w_n is transformed into a vector $u_n = [r^w; r^{ch}]$ that includes double sub-vectors: the word-based embedding $r^w \in \mathbb{R}^{d^w}$ and the character-based embedding $\mathbb{R}^{cl_u}$ of w_n. Word-based embeddings are used for capturing semantic and syntactic features, while character-based embeddings are for morphological information.

3.1.1 Word-level representation

To create word embeddings, the column vectors encode word-based embeddings in an embedding matrix $E^w \in \mathbb{R}^{d^w \times |V^w|}$. Each column $E_i^w \in \mathbb{R}^{d^w}$ denotes the word-based embedding of the *i*-th word in the vocab-table. Then, the matrix-vector product is used to convert word w into its word-based embedding r^w:

$$r^w = E^{w_v w}, \tag{1}$$

where υ^w is a vector of size is $|V^w|$ with a value of 1 at index υ and 0 in the remaining spot. The matrix E^w is a parameter used for learning, and the word-based embedding's size is a hyperparameter that is tuned in terms of specific requirements.

3.1.2 Character-level representation

In MRL, POS tagging also suffers from OOV problem, and its training data with syntactic annotation is very scarce. Many research studies have revealed that the optimal solution to tackle this issue is to perform word-level representation from a large set of unlabeled data, such that the syntactic and semantic similarities of the words can be better modeled using word-level representation. It is a fact that word embedding with more vocabulary can ameliorate the OOV problem, but it does not capture the morpheme-based information from the character n-grams of a word which would be a useful feature to indicate word form in POS tagging of highly inflected languages. For example, In the Uzbek POS tagging, informative features may occur at the start (such as the prefix "*na*" in "*natunush*", which means strangeness in English), or at the end (the suffix "*chi*" in "*oqutquchi*" means teacher in English), and those features can lead to many OOVs and data sparsity.

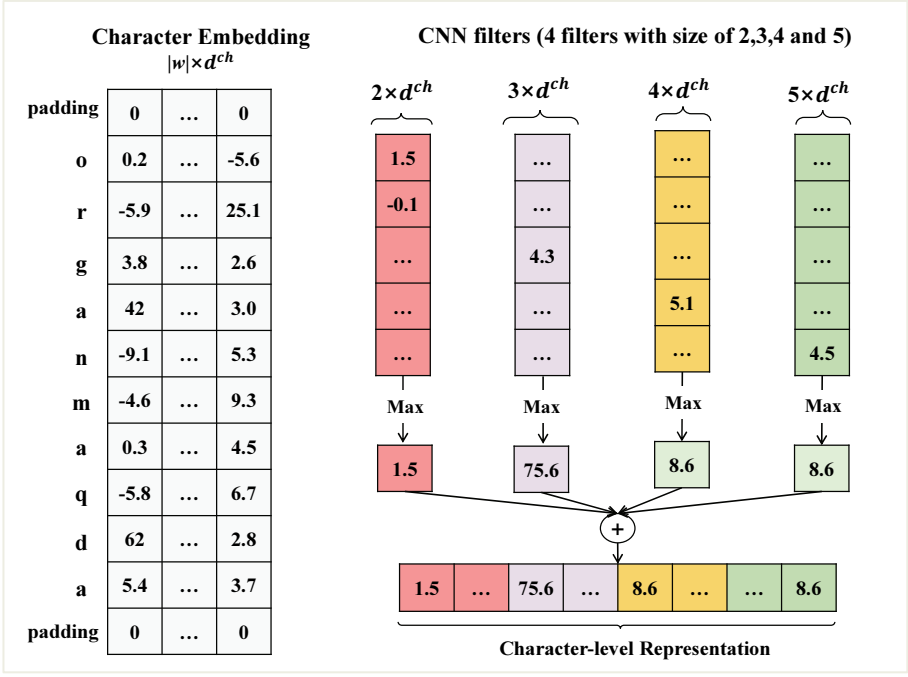


Fig. 1. The proposed CharNN architecture for character-level representation.

To compensate for these issues, we propose to augment the word-based representation with character-level features and adapt a CNN-based word embedding model to get a more precise character-level feature for each word, as illustrated in Fig. 1. Model details are as follows:

- The first layer is an embedding layer with a size equal to vocab-size $|w|$ and a dimension of 32 for each character. The purpose of this layer is to encode similarities between character into a multi-dimensional space in which two characters appearing in similar context are near each other.

- The second layer is composed of four different convolutional kernels with sizes of 2,3,4 and 5, and each one has 25 dimensions. They overlap along the different kernels and can encode pattern of 2,3,4 and 5 consecutive character n-grams. We expect those kernels to gain morphemes of up to 5 characters.
- In the output layer, each of convolutional filters is fed through a max pooling layer (size of 4, one for each filter), and the pooling results are concatenated to reduce the convolution results into a single embedding vector with sizes of 100.

Given a word ω which is composed of M characters $\{c_1, c_2, \dots, c_M\}$, first we convert every character c_m into a character embedding r_m^{ch}. Like the word-based embedding, the column vectors encode character-based embeddings in an embedding matrix $E^{ch} \in \mathbb{R}^{d^{ch} \times |V^{ch}|}$. Given a character c, then the corresponding embedding r^{ch} is formulated based on the matrix vector product:

$$r^{ch} = E^{ch} v^c, \tag{2}$$

where v^c is a vector size of $|V^{ch}|$ with a value of 1 at index c and 0 in the remaining positions. Then, the sequence of character embeddings $\{r_1^{ch}, r_2^{ch}, \dots, r_M^{ch}\}$ is an input for the convolution layer, which performs a matrix-vector procedure on every window size k^{ch} of character n-grams in the char sequences $\{r_1^{ch}, r_2^{ch}, \dots, r_M^{ch}\}$. The concatenation of character embedding m is defined as a vector $z_m \in \mathbb{R}^{d^{ch} k^{ch}}$.

$$z_m = \left(r^{ch}_{m-(k^{ch}-1)/2}, \dots, r^{ch}_{m+(k^{ch}-1)/2}\right), \tag{3}$$

Different convolution layers perform the computation of the i-th element of the vector r^{w_ch}, which is the character-based representation of a given word.

$$[r^{w\_ch}]_i = \max_{1<m<M} [E^0 z_m + b^0]_i, \tag{4}$$

where $E^0 \in \mathbb{R}^{cl_u \times d^{ch} k^{ch}}$ denotes the weight matrix of convolutional layer. A similar matrix is exploited to obtain the local features of each character n-grams of a word. Finally, the max-over-time pooling is performed to attain a global-feature vector for a word. Matrices E^{ch} and E^0, and vectors b^0 are parameters that our model needs to learn. The hyper-parameters are selected based on a specific task and include the character vector size d^{ch}, the number of convolution units cl_u, and the window size k^{ch}.

3.2 Scoring and Tag Inference

In the scoring model, we adopt the sliding window scheme [17] to assign all tags T for each word a score and follow the viewpoint that the tag of a word is generally determined by its surrounding words, especially which is viable in many NLP tasks, like NER and Dependency Parsing. A sentence with N words $\{w_1, w_2, \dots, w_N\}$, which is already transformed into the joint embedding: word-wise and character-wise $\{u_1, u_2, \dots, u_N\}$. To perform the tag scores prediction for the n-th word in a phrase, a vector x_n is generated from the concatenation of embeddings k^w and centered on the n-th word:

$$x_n = \left(u_{n-(k^w-1)/2}, \dots, u_{n+(k^w-1)/2}\right)^T. \tag{5}$$

We employ special padding marks and tag words with indices that extend beyond the phrase boundary. The vector x_n is then passed through two layers of a neural network, which extract high level representations and perform the tag scoring:

$$s(x_n) = E^2 h(E^1 x_n + b^1) + b^2, \tag{6}$$

where matrices $E^1 \in \mathbb{R}^{hl_u \times k^w (d^w + cl_u)}$ and $E^2 \in \mathbb{R}^{|T| \times hl_u}$, and vectors $b^1 \in \mathbb{R}^{hl_u}$ and $b^2 \in \mathbb{R}^{|T|}$ are also parameters that our model learns, and h(.) is the hyperbolic tangent as a transfer function. Additionally, the context window size k^w and the number of hidden units hl_u are used as hyper-parameters.

In POS tagging, there is a strong correlation between the adjacent words and their tags. Some of them are seen in chunks (e.g., name entities with two or more-word tokens), and the parts of them are less likely to occur behind the other tags (especially, adjectives do not follow verbs in SOV languages). Therefore, a sentence-level tag prediction scheme that focus more on the structural information of word sequence can perform better with tag dependencies. For a given list of words, this approach uses a transition score $A_{t,u}$ when transitioning from tag $t \in$T to $u \in$T, and a score $A_{0,t}$ when starting at the t-th tag. Given the expression $[w]_1^N = \{w_1, w_2, \dots, w_N\}$, scoring a tag path $[t]_1^N = \{t_1, t_2, \dots, t_N\}$ is defined in the following way:

$$S([w]_1^N, [t]_1^N, \theta) = \sum_{n=1}^{N} \left(A_{t_{n-1}, t_n} + s(x_n)_{t_n} \right), \tag{7}$$

where $s(x_n)_{t_n}$ is the score of tag t_n at word w_n, and θ is all the trainable parameters $(E^w, E^{ch}, E^0, b^0, E^1, b^1, E^2, b^2, A)$. Then, we use the Viterbi algorithm and predict the final sentence tags $[t^*]_1^N$ that can lead to the maximal score:

$$[t^*]_1^N = \underset{[t]_1^N \in T^N}{arg\,max}\, S([w]_1^N, [t]_1^N, \theta). \tag{8}$$

3.3 Model Training

The network is trained using the same method presented in [17], namely, minimizing the negative log likelihood on the training dataset D. In this method, we model the sentence score as a path conditional probability and calculate the exponentiation score, then normalize all possible paths that have high probability. Taking the log, we can come to the following formulation in terms of conditional log-probability:

$$log\, p([w]_1^N, [t]_1^N, \theta) = S([w]_1^N, [t]_1^N, \theta) - log \left(\sum_{\forall [u]_1^N \in T^N} e^{S([w]_1^N, [t]_1^N, \theta)} \right). \tag{9}$$

Then, the stochastic gradient descent (SGD) is used to degrade the negative log-likelihood regarding θ in the following way:

$$\theta \mapsto \sum_{([w]_1^N, [y]_1^N, \theta) \in D} - log\, p\,([y]_1^N, [w]_1^N, \theta), \tag{10}$$

where $[w]_1^N$ represents a sentence from training dataset D, and $[y]_1^N$ is the tag. The proposed network architecture can be effectively computed using the backpropagation technique.

4. Experiments

In the following part, we demonstrate our experimental settings and performance results of applying the suggested CharNN architecture to the POS tagging of morphologically rich languages such as Uyghur, Uzbek, and Kyrgyz.

4.1 Pre-processing

There are no available normalized text corpora that those three MRL involved. Considering the quality and scale of the corpus, the news corpora of Uyghur, Uzbek, and Kyrgyz languages provided by Leipzig University (http://corpora.un-leipzig.de) are used, as the source of unlabeled monolingual raw text. These raw texts are preprocessed based on the following step: (1) removing sections that are not in specific languages; (2) extracting a stem from a raw word in the text; (3) and removing sentences including less than two words. Intending to investigate the importance of word-stem on word-based representation, the words are transformed into their word-stem form by removing the inflected parts, and the word-stem version of corpus are stored for later use.

In the word embeddings of three MRL, the same parameters for those three languages are selected, except for the minimum word frequency (MWF). As various word formation and distribution of three languages in the corpus lead do different word capacity that can reflect the language characteristics. in Uyghur, the frequency of occurrence of a word must be more than 10 to be contained in the vocabulary, which produces a total of 32,642 entries. For Uzbek, MWF with 7 is selected and total 30,561 entries are generated. For Kyrgyz, MWF of 7 is also considered, which leads to 26,548 entries in total. It is noted that the suffixes will not be included, as the MWF is done solely on the word-stem. This is another way of resolving the data sparsity and OOVs problem in MRL that are highly agglutinative and have scarce annotated data. For the experiments on three MRL, the fastText 100-dimensional embedding vector is trained for each language.

In the character embeddings, the size of the character vocabulary of three MRL is too limited in comparison with the word vocabulary in the corpora. Hence, the training corpora for POS tagging are large enough to effectively train character-based embeddings. It is vital that the embeddings based on characters must strictly use raw (words with a root and suffixes) words. Hence, the network could capture the morphological information and word form of a word.

4.2 Dataset

Since those languages lacks the publicly available annotated dataset for POS tagging, the custom datasets (i.e., MultiPOS_ukg) for Uyghur, Uzbek, and Kyrgyz, which include 12 POS tags are created using our tokenizer and annotated with help of language specialist. It will be released online soon. The specifics about datasets are provided in Table 1.

To make our results more concrete and comparable to other relevant research on POS tagging, the proposed approach is additionally tested on a publicly released Turkish dependency parsing dataset. It is because Turkish is also highly inflected language that it is topologically very similar to those three MRL, and they share the same grammatical structure in sentence. The additional POS tagging dataset for Turkish is METU Turkish Treebank [18] (http://tools.nlp.itu.edu.tr/Datasets), which consists of 56k

word tokens, 5,600 sentences, and 13 POS tags. in Turkish POS tagging part, the same training configuration and data partition are used, as in those three MRL.

Table 1. Datasets for Uyghur, Uzbek, and Kyrgyz POS tagging

Language	Set	Number of sentences	Number of tokens (distinct stem)
Uyghur	Training	20,000	28,325
	Development	1,000	1,376
	Test	500	712
Uzbek	Training	20,000	22,159
	Development	1,000	1,102
	Test	500	532
Kyrgyz	Training	15,000	25,942
	Development	200	335
	Test	500	844

4.3 Experimental Setup

4.3.1 Hyper-parameter setting

The development set is used to tune the network hyper-parameters, since various hyper-parameter combinations can produce almost the same consequences. Learning rate (LR) is typically one of the hyper-parameters having a substantial impact on the prediction process in SGD training. Therefore, our focus on tuning the parameters is mostly about LR, in comparison with others. However, when using the same number of training iterations, the results from LR values of 0.005 to 0.01 are very close. Additionally, a learning rate setting is used to reduce LR r based on the training epoch t. So, the LR for each epoch r_t is computed in the following way:

$$r_t = \frac{r}{t}.$$

The LR setting and its generalization are presented in [17]. Noting that the same hyper-parameters are used for Uyghur, Uzbek, Kyrgyz and Turkish, which could indicate that our suggested approach is robust to multiple languages which are topologically identical. The selected hyper-parameters settings are presented in Table 2.

4.3.2 Baseline setup

Extensive experiments on two datasets, including the MultiPOS_ukg and the METU Turkish Treebank, are conducted to validate the performance of the proposed CharNN. They include four morphologically rich languages. In our experiments, we test the proposed CharNN against several different baselines. The first baseline is word-based neural network (WordNN) that uses only randomly initialized word-level representation. The second baseline is WordNN that only includes suffix embedding in which size of 2, 3, 4, and 5 suffixes are used. The third baseline is also WordNN that is fed with word-stem-level representation instead of word. the fourth baseline is WordNN that includes two additional handcrafted features: word-stem and suffix. In addition, to ensure a fair comparison with other related research work, our proposed model is applied to a Turkish dataset with the same features and model hyper-parameters for the POS tagging task.

Further investigation suggests that integrating the character-based representation with word-based models could be as they capture orthographic, morpho-syntactic, and semantic components of word similarity.

Table 2. Hyper-parameter settings

Component	Value
d^w (Word) Dim	100
k^w Word Count. Window	5
d^{ch} (Char) Dim	32
k^{ch} Char. Count. Window	[2, 3, 4, 5]
cl_u Convol. Units	100
hl_u Hidden Units	50
r Learning Rate	0.0075

4.4 Results and Analysis

POS tagging is a classification task where each word in a sentence will be tagged with the appropriate syntactic category based on its usage. In this work, the tagging performance of different experimental runs are reported using accuracy (Acc) metric.

In the first set of experiments, the proposed CharNN is evaluated alongside different baseline models using our custom dataset MultiPOS_ukg. The performance results are presented in Table 3, where stem denoting additional *word-stem* feature, suf. denoting suffix features, and OOTV indicating out-of-the-tagged-vocabulary.

Table 3. Tagging accuracies of different NN's in POS tagging on the Uyghur, Uzbek, and Kyrgyz dataset (MultiPOS_ukg) (unit: %)

Method	Feature	Uyghur		Uzbek		Kyrgyz	
		Acc	Acc OOTV	Acc	Acc OOTV	Acc	Acc OOTV
CharNN	N/A	90.47	85.48	80.16	73.74	79.32	80.68
WordNN	Suf.+Stem	76.89	86.85	78.32	71.31	75.21	79.48
WordNN	Stem	75.17	79.95	76.98	68.45	73.08	76.05
WordNN	Suf.	73.35	80.61	75.79	63.64	71.34	76.96
WordNN	N/A	73.19	72.92	75.68	62.40	70.13	71.94

As seen in Table 3, the WordNN that does not use character-level embeddings and has no additional features consistently produces low accuracy across all languages, including OOTV tagging. This is due to the fact that word-level embedding can only learn semantic information about a word and ignores the word-form information, which is crucial for a word in MRL. This form of word-level representation is ideal for languages like Chinese and English whose grammatical structures considerably diverge from those of MRL. Therefore, simply using word-level embeddings performs poorly on MRL scenario.

Combining stemming and suffix features substantially improves the accuracy (by up to 3, 3, and 5 points in Uyghur, Uzbek, and Kyrgyz, respectively) and OOTV accuracy (by up to 14, 9, and 8 points in Uyghur, Uzbek, and Kyrgyz, respectively), when WordNN are also utilized. It is found that additional features significantly boost the performance of OOTV by a much larger margin than general tagging.

This suggests that stemming and suffix features can assist WordNN in modeling morphological information from the word, even if they lack word embeddings.

It is worth noting that the CharNN model nearly outperforms the WordNN model on different experiment runs. This seems to show that character-based representation learning is more effective solution to the data sparsity problem than the WordNN. Additionally, it is confirmed that there are a few words that are OOTV if they do not appear in the training set. The CharNN also shows its flexibility in processing sub-word information of a word, using different kernels to capture n-grams of varying lengths. In our setting, a kernel with a minimum length of 2 is enough for capturing small morphemes that are possibly a prefix or suffix, whereas a kernel with a maximum length of 6 is capable of capturing typical words. the CharNN always outperforms the word-based model due to its ability to encode patterns ranging from morphemes up to words.

Another interesting case is found in Table 3 that there are discrepancies in performance amongst the three languages. In Uyghur POS tagging, using the CharNN model improves the accuracy from 84.89% to 88.47%, while the performance of OOTV did not have a significant impact. Since a Uyghur word is highly inflectional and has considerable complex grammatical form. Instead, there are fairly less grammatical changes in words from Uzbek and Kyrgyz so that those handcrafted features show relatively small improvement in the POS tagging for OOTV words. Overall, it can be argued that CharNN is more suitable for processing the morphemes in highly agglutinative languages with the help of convolutional kernels.

4.4.1 Comparison with SOTA taggers for METU Turkish Dataset

In this experiment, the effectiveness of the proposed CharNN architecture on Turkish dataset is investigated, and the resulting performance is compared to that of previous research works. the results are presented in Table 4.

Table 4. Comparison with related study on the Turkish META-Sabancı Turkish Treebank dataset

Study	Model	Accuracy (%)
This work	Char-based CNN	90.36
Esref and Can [18], 2019	CRF	89.20
Bhavan et al. [10], 2018	RNN	79.70
	LSTM	89.00

The performance show that the CharNN model outperforms the other models with a slight advantage, where ours is 1.36% higher than the LSTM, 1.16% higher than the CRF, and 10.66% higher than the RNN. Moreover, the CRF and LSTM are very close and boost the RNN by a large margin. CRF simply has a much smaller context windows and requires complex feature engineering that are hard to acquire and time-consuming, thus the slightly poor performance. The LSTM theoretically can be capable of encoding long-distance context. However, it is constructed incrementally with repetition. This DNN architecture would take more data to learn the same sequence than CNN, which can directly match the pattern by employing a large kernel despite the lack of annotated POS data. It is conjectured that the CharNN's superior performance is due to the fact that it can more readily capture morpho-syntactic information of a word that competing model. In our setting, the CharNN uses four different kernels with

sizes of 2, 3, 4, and 5 which can capture morphemes with different lengths or even a normal word, without depending on any extra handcrafted features.

4.4.2 Performance comparison on non-agglutinative language dataset

Extensive experimental findings indicates that the proposed CharNN method appears to significantly enhance the performance of POS tagging for MRL. Considering the performance of the CharNN in non-agglutinative languages, the effectiveness of different representation methods which include word-level, character-level, and joint representation are experimented on the Chinese and English datasets, as a further investigation. Our purpose is to investigate CharNN's generalizability and to evaluate how much it relies on training languages. The selected languages are Uyghur, Uzbek, and Kyrgyz as MRL, and the counterexample languages like Chinese and English. The MRL practically has large numbers of grammatical cases which result from different suffixes of varying lengths. In Fig. 2, join denotes the joint model, char denotes a model trained only with CharNN, and word is a model trained only with WordNN.

In Fig. 2, it is demonstrated that the char model is superior to the word model in complex morphological scenarios. Uyghur shows significantly better improvements than Uzbek and Kyrgyz in both the character-based and word-based models in MRL. Since, a Uyghur word consists of a complex variation of vowels, consonants, and grammatical features in comparison with Uzbek and Kyrgyz that have relatively weaker inflection. On the English and Chinese dataset, however, the word-based and join models provide a slightly better outcomes than the character-based model. According to these empirical findings, character-level representations are more significant for complex morphological languages due to their capability to encode the morphological information more effectively.

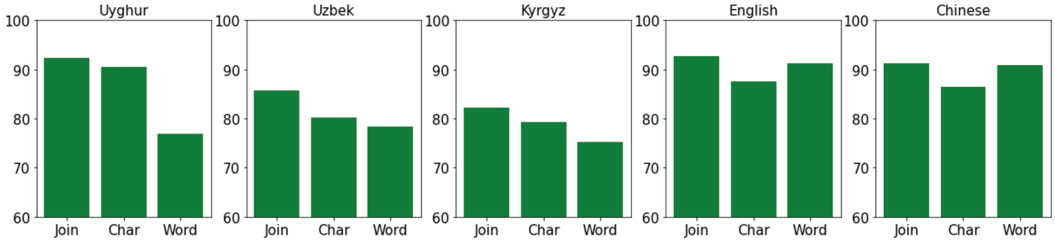


Fig. 2. Evaluation results on non-agglutinative language datasets. Uyghur, Uzbek, and Kyrgyz use our custom dataset MultiPOS_ukg and zh_GSD (https://github.com/UniversalDependencies/UD_Chinese) for Chinese, En_EWT (https://github.com/UniversalDependencies/UD_English) for English.

4.4.3 Analysis on the OOV

In this work, the reason for utilizing a character-based representation model is based on the idea that it can tackle the OOV issues. In order to prove the hypothesis, an ablation experiment is conducted on the unnormalized text, which may include intentional or accidental misspelled words, namely, OOVs. While there is no social media data that exists many misspelled words for those three languages. However, an experiment is designed to imitate the randomly edited text by manually changing the word in the development set using insertion, deletion, substation, and swap. For instance, in a word "*gulum*" if the position 2 (0-based) is modified, the resulting words are "*guxlum*," "*guum*," "*guxum*," and "*guulm*," where *x* is a random character from the language's alphabet.

For each operation, a group of changed development sets are constructed, and the words that are of minimal length 2 are edited with a probability of 0.25, 0.5, 0.75, or 1 depending on the operation. In the experiment, the normal training sets are used for training while the modified development-sets are used to predict POS tag. The average accuracies are illustrated in Fig. 3.

As seen in the Fig. 3, all models suffer from a growing percentage of artificially made OOV words, but CharNN always degrades the least and steady. This suggested that CharNN is more robust to misspelled words. When reviewing the individual examples of misspelling, CharNN is less sensitive to replacement, while insertion and deletion yields strong impact and swap has the most negative impacts on its performance. In the substitution, the distortion to the character n-grams is smaller based on different lengths and this leads to smaller negative impact on insertion and deletion. However, in the case of swap, it is shown the same effect which is more like substituting two chars instead of one, thus larger degradation.

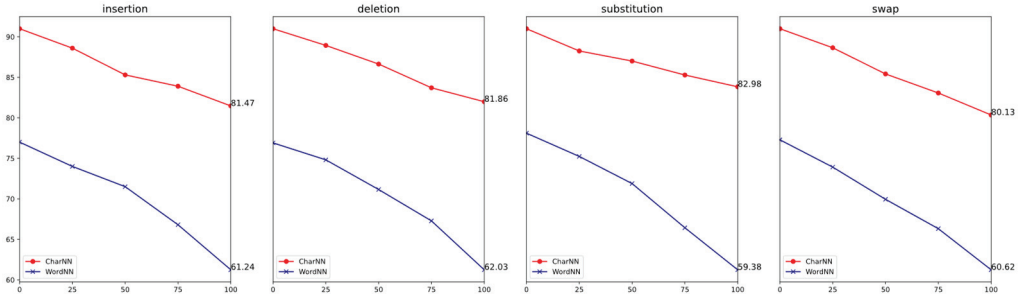


Fig. 3. Average POS tagging accuracies on the unnormalized development sets with the four adjustments.

5. Conclusion

In this study, a Character-based Neural Network Architecture (CharNN) for POS tagging task in three MRL is proposed. The proposed architecture uses the character-based representation to allow effective feature extraction from word of any size and combines it with word-level representation. Moreover, Regarding the lack of publicly available POS tagging dataset in those three languages, the custom datasets (i.e., MultiPOS_ukg) are constructed. The CharNN is experimented on the MuliPOS_ukg set and non-agglutinative language datasets. The experimental performances on the MultiPOS_ukg set show that the proposed CharNN can improve POS tagging accuracy by large margin than the word-based model in each language, with an improvement of 14, 2, and 4 points for Uyghur, Uzbek, and Kyrgyz respectively. Further, to assess the robustness of our model against the OOV problem in POS tagging, the proposed model is performed on the manually unnormalized development-set which is generated by editing the words in the normal development-sets. The performance of the CharNN degrades much slower than other model and it can be verified that the proposed model is more robust against the unnormalized case.

In this work, the more focus is on the character representation, which mainly aims at exploring the efficacy of CharNN in dealing with the OOV words in POS tagging. However, the tag scoring and inference is our limitation, and it could potentially further improve the POS tagging performance. This part is left for the future work.

In addition, we intend to examine the extent to which Neural Machine Translation (NMT) system for Chinese and minority languages can benefit from this proposed architecture.

Acknowledgement

This work is supported by the Doctoral Fund of Xinjiang Normal University (No. XJNUBS2007).

References

[1] X. Yu and N. T. Vu, "Character composition model with convolutional neural networks for dependency parsing on morphologically rich languages," in Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL), Vancouver, Canada, 2017, pp. 672-678.

[2] T. V. Ngo, T. L. Ha, P. T. Nguyen, and L. M. Nguyen, "Overcoming the rare word problem for low-resource language pairs in neural machine translation," 2019 [Online]. Available: https://arxiv.org/abs/1910.03467.

[3] M. Grohe, "word2vec, node2vec, graph2vec, x2vec: towards a theory of vector embeddings of structured data," in *Proceedings of the 39th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, Portland, OR, 2020, pp. 1-16.

[4] C. I. Eke, A. Norman, L. Shuib, F. B. Fatokun, and I. Omame, "The significance of global vectors representation in sarcasm analysis," in *Proceedings of 2020 International Conference in Mathematics, Computer Engineering and Computer Science (ICMCECS)*, Ayobo, Nigeria, 2020, pp. 1-7.

[5] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 135-146, 2017.

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

[7] R. D. Deshmukh and A. Kiwelekar, "Deep learning techniques for part of speech tagging by natural language processing," in *Proceedings of 2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*, Bangalore, India, 2020, pp. 76-81.

[8] K. K. Akhil, R. Rajimol, and V. S. Anoop, "Parts-of-Speech tagging for Malayalam using deep learning techniques," *International Journal of Information Technology*, vol. 12, pp. 741-748, 2020.

[9] W. Khan, A. Daud, K. Khan, J. A. Nasir, M. Basheri, N. Aljohani, and F. S. Alotaibi, "Part of speech tagging in Urdu: comparison of machine and deep learning approaches," *IEEE Access*, vol. 7, pp. 38918-38936, 2019.

[10] C. A. Bahcevan, E. Kutlu, and T. Yildiz, "Deep neural network architecture for part-of-speech tagging for Turkish language," in *Proceedings of 2018 3rd International Conference on Computer Science and Engineering (UBMK)*, Sarajevo, Bosnia and Herzegovina, 2018, pp. 235-238.

[11] K. K. Todi, P. Mishra, and D. M. Sharma, "Building a Kannada POS tagger using machine learning and neural network models," 2018 [Online]. Available: https://arxiv.org/abs/1808.03175.

[12] S. Kumar, M. A. Kumar, and K. P. Soman, "Deep learning based part-of-speech tagging for Malayalam Twitter data (Special issue: deep learning techniques for natural language processing)," *Journal of Intelligent Systems*, vol. 28, no. 3, pp. 423-435, 2019.

[13] M. Maimaiti, A. Wumaier, K. Abiderexiti, and T. Yibulayin, "Bidirectional long short-term memory network with a conditional random field layer for Uyghur part-of-speech tagging," *Information*, vol. 8, no. 4, article no. 157, 2017. https://doi.org/10.3390/info8040157

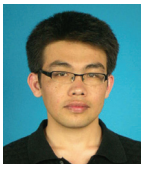
[14] S. Fouladi, M. J. Ebadi, A. A. Safaei, M. Y. Bajuri, and A. Ahmadian, "Efficient deep neural networks for classification of COVID-19 based on CT images: virtualization via software defined radio," *Computer Communications*, vol. 176, pp. 234-248, 2021.

[15] A. Javaheri, N. Moghadamnejad, H. Keshavarz, E. Javaheri, C. Dobbins, E. Momeni-Ortner, and R. Rawassizadeh, "Public vs media opinion on robots and their evolution over recent years," *CCF Transactions on Pervasive Computing and Interaction*, *2*, 189-205, 2020.

[16] S. Fouladi, A. A. Safaei, N. Mammone, F. Ghaderi, and M. J. Ebadi, "Efficient deep neural networks for classification of Alzheimer's disease and mild cognitive impairment from scalp EEG recordings," *Cognitive Computation*, vol. 14, pp. 1247-1268, 2022.

[17] R. Collobert, "Deep learning for efficient discriminative parsing," in *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS)*, Fort Lauderdale, FL, 2011, pp. 224-232.

[18] Y. Esref and B. Can, "Using morpheme-level attention mechanism for Turkish sequence labelling," in *Proceedings of 2019 27th Signal Processing and Communications Applications Conference (SIU)*, Sivas, Turkey, 2019, pp. 1-4.



Samat Ali **https://orcid.org/0000-0002-9425-1980**

He received B.S. and M.S. degrees in School of Computer Science and Engineering from Xinjiang University in 2006 and 2010, respectively. Since September 2012, he is with the School of Computer Science and Engineering from Xinjiang Normal University as a Teaching Assistant. His current research interests include Natural Language Processing and Machine Translation.

Alim Murat **https://orcid.org/0000-0001-8510-7808**

He received B.E. and M.S. degrees in School of Computer Science and Technology from Xinjiang Normal University in 2011 and 2014, respectively. He finished his Ph.D. degree in Computer Science from Xinjiang Technical Institute of Physics & Chemistry, Chinese Academy of Science in June 2017. Since July 2017, he has been working in the Xinjiang Normal University as a lecture. His current research focus includes natural language processing and Machine Translation.