Neural Joint Model for Part-of-Speech Tagging and Entity Extraction

Wazir Ali
aliwazirjam@gmail.com
School of Computer Science and Engineering, University of Electronics Science and Technology of China
Chengdu, Sichuan, China

Rajesh Kumar
rajakumarlohano@gmail.com
School of Computer Science and Engineering, University of Electronics Science and Technology of China
Chengdu, Sichuan, China

Yong Dai
daiyongya@yahoo.com
School of Computer Science and Engineering, University of Electronics Science and Technology of China
Chengdu, Sichuan, China

Jay Kumar
jay@std.uestc.edu.cn
School of Computer Science and Engineering, University of Electronics Science and Technology of China
Chengdu, Sichuan, China

Saifullah Tumrani
saif.tumrani@std.uestc.edu.cn
School of Computer Science and Engineering, University of Electronics Science and Technology of China
Chengdu, Sichuan, China

ABSTRACT

Part-of-speech tagging and named entity recognition (NER) are fundamental sequential labeling tasks in natural language processing (NLP), where joint learning of both tasks is an effective one-step solution. Limited efforts have been made by existing research to meet such needs for Sindhi language. As POS tagging and NER are highly correlative sequence tagging tasks, so most often, a word recognized by the NER system may be recognized as a noun by a POS tagger. Thus, in this paper, we propose a neural joint model based on a bidirectional long-short term memory (BiLSTM) network and adversarial transfer learning to incorporate syntactic information from two tasks by using task-shared information. The syntactic structure captures and provides the information of long-range dependencies among words. Moreover, the self-attention is employed to capture intra-sentence dependencies to the joint model explicitly. Empirical results on two benchmark datasets show that our proposed joint model consistently and significantly surpass the existing methods.

CCS CONCEPTS

• **Computing methodologies** → **Natural language processing**; *Neural networks.*

KEYWORDS

Adversarial transfer learning, Parts-of-speech tagging, Named Entity Recognition, Sindhi language

ACM Reference Format:
Wazir Ali, Rajesh Kumar, Yong Dai, Jay Kumar, and Saifullah Tumrani. 2021. Neural Joint Model for Part-of-Speech Tagging and Entity Extraction. In *2021 13th International Conference on Machine Learning and Computing (ICMLC '21), February 26-March 1, 2021, Shenzhen, China.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3457682.3457718

ICMLC '21, February 26-March 1, 2021, Shenzhen, China

ACM ISBN 978-1-4503-8931-0/21/02…$15.00
https://doi.org/10.1145/3457682.3457718

1 INTRODUCTION

Part-of-speech tagging and NER have been two fundamental tasks for Sindhi NLP [2, 4, 8, 43]. The former one assigns a POS tag to each word to indicate its syntactic property in the sentence. The latter determines named entity (NE) boundaries and classifies them into pre-defined categories on the top of POS tagging results. State-of-the-art methods treat both tasks as structural learning problems, using either transition-based incremental models for sequence tagging models [14, 37, 53]. The former can exploit dynamic decoding, while the latter is flexible in feature engineering. As POS tagging and NER are highly correlative sequential labeling tasks, so most often, a word recognized by the NER system may be recognized as a noun by a POS tagger. Thus, both tasks are often selected as the target tasks at the same time. For example, an end-to-end sequence labeling model [42] that automatically learns from word-level and character-level representations and achieved state-of-the-art performance in both tasks. Recently, joint models that perform in a single model have achieved better performances due to making use of mutual interaction between the two tasks and their capabilities of reducing error propagation [12, 23, 56]. Among the work of joint POS tagging and NER [52] is one of the representative methods because of its high efficiency and simplicity. Therefore, it would be a good way to design a joint neural model for Sindhi NER using POS tagging.

Little work exists to tackle the POS tagging and NER tasks in Sindhi language, and some issues still have not been well addressed. One significant drawback is that there is only a minimal amount of annotated data available. A recently proposed small annotated dataset [18] consists of 6,842 tokens. More recently, a benchmark POS tagset [4] has been introduced by annotating the news corpus [5] of Awami-Awaz and Kawish Sindhi newspapers using the Doccano text annotation tool. Moreover, the SiNER [8] is also a recently proposed gold-standard dataset of Sindhi NER. Moreover, most proposed neural models cannot explicitly capture long-range dependencies when predicting NE type, and the context information is essential to determine such NE types. The BiLSTM [48] can learn

Task	عدنان ميندريس جيڪو ترڪي جو وزير اعظم رهيو Rahyo azam vazeere jo Turkey jaiko Menderes Adhan Adnan Menders who remained prime minister of Turkey							
SPOS	رهيو	اعظم	وزير	جو	ترڪي	جيڪو	ميندريس	عدنان
	VB	NN	NN	DET	NNP	DET	NNP	NNP
SNER	رهيو	اعظم	وزير	جو	ترڪي	جيڪو	ميندريس	عدنان
	O	I-TITLE	B-TITLE	O	B-GPE	O	I-PERSON	B-PERSON

Figure 1: An example of the similarities between Sindhi POS tagging and NER

long-range dependencies to capture contextual information from the given input. Thus, we employ the BiLSTM encoder, adversarial transfer learning, and self-attention for the joint modeling of both Sindhi POS tagging and NER tasks to incorporate context features, syntactic and dependency relations using task-shared information.

As shown in Figure 1, given a sentence (*Adnan Menders who remained prime minister of Turkey*), POS tagging and NER tasks have the same word boundaries such as (Adnan) and (Menders), while boundaries in Sindhi NER are more coarse-grained than POS tagging task for certain words. For example, a NER system takes (Adnan Menders) as a whole NE, and the POS tagging task will split (Adnan) and (Menders). In order to incorporate such contextual, syntactic, and word boundary information from two tasks, the proposed neural model jointly performs Sindhi POS tagging with NER tasks, respectively.

Sindhi POS tagging has been previously investigated in various writing scripts of including Devanagari [45], Roman [49], and Persian-Arabic [18, 43, 44, 51]. Earlier work on the Sindhi POS tagging has been carried out by using the rule-based [43] and lexical-based [44], support vector machine, [51], and CRF [45] on a very limited amount of data. While the NER task was initially coined [3] by highlighting the importance and challenges related to the development of Sindhi NER system. Later, rule-based [24, 31, 46] methods were introduced and evaluated on a small amount of data lack open-source implementation. These rule-based systems have two main deficiencies of high development cost and the continuous maintenance of assigned rules in case of the addition of new words or new entities in the language. Most recently, a large gold-standard dataset [8] is proposed for SiNER, which is a sophisticated addition in the resources for Sindhi natural language processing (SNLP). As a basic task in the NLP area, the performance is not satisfactory. Fortunately, the amount of supervised training data for Sindhi POS tagging and NER is available. Both tasks share many similarities, such as word boundaries, and an NE recognized by the NER system may be recognized as a noun by the POS tagger. Such co-related features share the same class space between Sindhi POS tagging and NER, which we call task-shared information. We utilize both SiPOS [4] and SiNER [8] datasets in the experimental setup. Our contributions are summarized as follows:

(1) We propose a joint neural model by incorporating task-shared POS tagging information into the Sindhi NER task. The proposed model is mainly based on BiLSTM encoder and adversarial transfer learning to integrate context features, syntactic information, and dependency relations using task-shared information. It is the first work to apply the adversarial transfer learning method to the Sindhi NER task.
(2) We incorporate a self-attention mechanism into our model, aiming to capture long-range dependencies to synthesize the hidden representations of the BiLSTM network.
(3) The experiments are conducted on two different Sindhi POS tagging and NER datasets. Our proposed joint model significantly and consistently outperforms the previous models.

2 RELATED WORK

POS: Generally, the POS tagging is treated as a sequence tagging problem and commonly solved by using transition-based or conditional random field (CRF) models [14, 33, 37, 39]. Recently, neural POS taggers [11, 16, 32, 47] have advanced state-of-the-art by performing well using word-level and character-level features. In Sindhi, both word-level and character-level features are important [6, 7] for training a neural model. Little work exists to tackle the POS tagging and NER tasks in Sindhi language. Sindhi parts-of-speech (SPOS) tagging has been previously investigated in various writing scripts, including Devanagari [45], Roman [49], and Persian-Arabic [18, 43, 44, 51]. Earlier work on Sindhi POS tagging has been carried out by using the rule-based [43], lexical-based [44], support vector machine (SVM) [51], and CRF [45] due to the scarcity of labeled data. Recently, an annotated Sindhi POS dataset [18] contain 6842 tokens, which is not sufficient to train a supervised neural classifier. More recently, a benchmark dataset [4] namely SiPOS has been introduced by annotating the news corpus [5] of Awami-Awaz and Kawish Sindhi newspapers, which contain more than 293K tokens.

NER: Early studies on NER exploit SVM [28], Markov models [9], CRFs [37], often heavily relying on feature engine-ering [58] as a joint identification and categorization of NEs, respectively. In recent years, recurrent neural network (RNN) [19, 27, 36] based models have been introduced to NER task [15, 27, 38, 42]. These models exploit BiLSTM [48] for feature extraction and feed them into the CRF decoder. Afterwards, the BiLSTM-CRF [27] network is widely exploited as the baseline. Sindhi is one of the low-resource language [29] which lacks basic language resources for mature computational processing. Little work exists to tackle the NER problem in Sindhi language. Initially, [2] highlighted the importance and challenges related to Sindhi NER. Later, [24] introduced the first Sindhi rule-based NER system consists of ten entity types and evaluated on 200k words. However, the dataset and the proposed system lack open-source implementation for further verification and extension. Afterwards, [46] proposed a rule-based approach to deal with the Sindhi NER task's ambiguities by using an indexing approach. However, their work lacks empirical results, which can signify the handling of ambiguous situations while developing an automatic NER system. Recently, [31] also addressed the Sindhi NER problem using a rule-based approach on a small number of 936 words. Their approach is also language-dependent and tested on a very small number of words, which are insufficient to develop a robust NER system. Moreover, the rule-based systems have two main deficiencies of high development cost and the continuous maintenance of

assigned rules in case of the addition of new entities to the language. Most recently, a large gold-standard SiNER dataset [8] is proposed for Sindhi NER. It contains 1,338 news articles and more than 1.35 million tokens collected from the news corpus. The annotation is performed using the Doccano text annotation tool using the begin-inside-outside (BIO) tagging scheme.

Self-attention: The self-attention [54] has been extensively used to capture long-range dependencies between input and output. More recently, self-attention has also been widely used in RNNs [26, 30] to learn token-level dependency, which captures the internal structure of a sentence in the NER task. The connection of RNN layers with self-attention is the best practice for performance gain. It directly computes the dependency [40] by ignoring the distance between tokens in joint modeling. Thus, self-attention is a useful approach for both local dependencies and long-distance between tokens.

Adversarial training: Adversarial training have achieved great success in NLP [20, 22, 57] for domain adaptation, multi-task learning [13, 41], crowd-sourcing learning [55], cross-lingual transfer learning [34], and a shared-private network [10] in domain separation. Similar to our proposed model [12] opted the adversarial training to train NER and word segmentation tasks jointly. Different from these works, we exploit the adversarial neural network to jointly train Sindhi POS tagging and NER tasks to extract task-shared information from the Sindhi POS tagging. To our knowledge, it is the first work to apply adversarial transfer learning to the Sindhi NER task.

3 THE PROPOSED MODEL

The architecture of our proposed joint neural model is illustrated in Figure 2 Sindhi POS tagging and NER following the word-level and character-level sequence tagging paradigm [7], where the input is a character sequence $\mathcal{X} = x_1 x_2 \cdots x_i \cdots x_l$ and the output is a sequence of joint labels $\mathcal{Y} = y_1 y_2 \cdots y_i \cdots y_l$. We propose a neural joint model based on adversarial transfer learning, which has the ability to learn task-shared contextual, syntactic, and word boundary information from the POS tagging task. Moreover, it filters specific POS information and explicitly captures the long-range dependencies between arbitrary two words or characters in a sentence. The proposed model mainly consists of five layers: (a) embedding layer, (b) feature extraction layer, (c) self-attention, (d) task-specific CRF, (e) task discriminator. Each part of the proposed model is illustrated in detail as under:

3.1 Embedding Layer

The first step of our proposed model is to map words and discrete characters into the distributed representations. We pretrain contextual embeddings by associating all surrounding features above with richer word context information, including POS tagging and NER tags. Since we utilize fastText [5, 21] to learn word-level and character-level representations. The words w_t and its corresponding characters $[c_1, c_2, \cdots, c_n]$ are projected onto their own representations, and concatenation of learned representations is input for the BiLSTM network. In this way, the BiLSTM captures joint contextual representations at character-level as well as word-level.

3.2 Feature Extraction Layer

The LSTM network [25] is an extension of RNN [19] proposed to solve vanishing and exploding gradient problems. The LSTM network only leverages information from the past, knowing nothing about the future information. We use the BiLSTM network in order to incorporate the information from both sides of the sequence to extract features. The process in the hidden state of BiLSTM can be expressed as follows:

$$\overrightarrow{h}_i = \overrightarrow{\text{LSTM}}\left(\overrightarrow{h}_{i-1}, x_i\right) \tag{1}$$

$$\overleftarrow{h}_i = \overleftarrow{\text{LSTM}}\left(\overleftarrow{h}_{i+1}, x_i\right) \tag{2}$$

$$h_i = \overrightarrow{h}_i \parallel \overleftarrow{h}_i \tag{3}$$

here $\overrightarrow{h}_i \in R^{d_h}$ and $\overleftarrow{h}_i \in R^{d_h}$ are the hidden states of the forward and backward LSTM at i^{th} position, respectively. $\parallel$ is concatenation operation.

As depicted in Figure 2, a shared private feature extractor, we assign a private and shared BiLSTM layers for the task $k \in \{NER, POS\}$. The private layer is employed to extract task-specific features, and the shared layer learns the task-shared syntactic features as well as word boundaries. Formally, for a given input sentence from the dataset of task k, the hidden representations of shared BiLSTM and private BiLSTM can be computed as:

$$z_i^k = \text{BiLSTM}\left(x_i^k, z_{i-1}^k; \theta_z\right) \tag{4}$$

$$h_i^k = \text{BiLSTM}\left(x_i^k, h_{i-1}^k; \theta_k\right) \tag{5}$$

here θ_z and θ_k are the parameters of shared and private BiLSTM of task k, respectively.

3.3 Self-attention

We add a token level multi-head self-attention layer above the $Word_{Char}$ encoder layer as a set of query Q, key K, and value V to capture the dependency of a whole sentence. The concatenated character-level and word-level representations are passed through forward $\overrightarrow{LSTM}$ and backward $\overleftarrow{LSTM}$ for the input to the self-attention layer where $H = h_1, h_2, \cdots, h_n$ denotes the output of private BiLSTM and $Z = z_1, z_2, \cdots, z_n$ is the output of shared BiLSTM network. The dot product attention can be precisely described as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V \tag{6}$$

here Q, K, V are the query, keys, and value matrices, and in our setting, $Q = K = V = H$.

Firstly, a multi-head self-attention projects the Q, K, V linearly h times with different linear projections. Afterwards, h projections perform the scaled dot-product attention in parallel, and the final results of attention are concatenated and once again projected to get the new representation. Formally, the mechanism can be illustrated as follows:

$$\text{head}_i = \text{Attention}\left(QW_i^Q, KW_i^K, VW_i^V\right) \tag{7}$$

$$H' = (\text{head}_i \|, \ldots, \|, \text{head}_h)\, W_o \tag{8}$$

where W_i^Q, W_i^K, W_i^V are trainable projection parameters and W_o is also a trainable parameter.

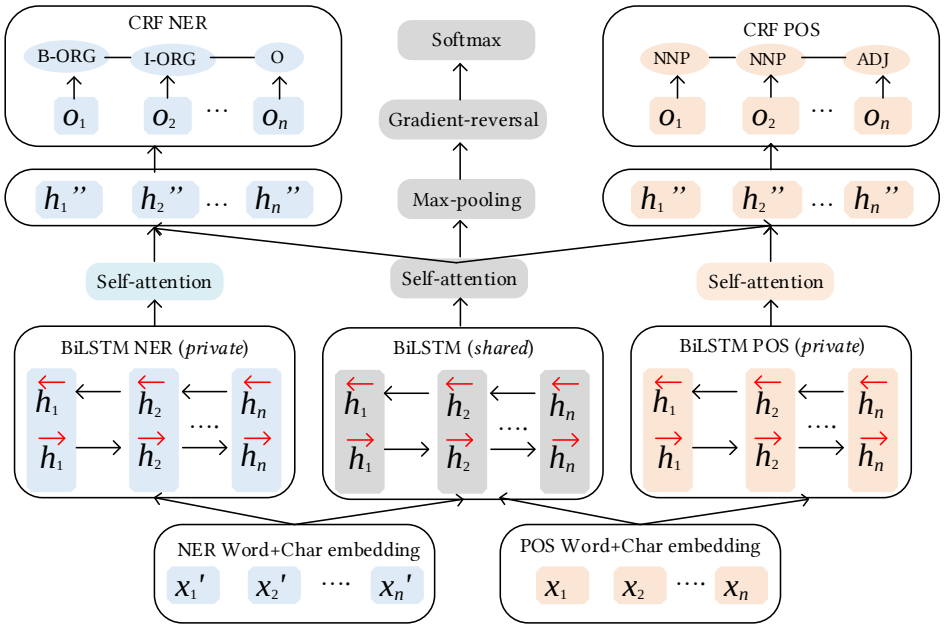

Figure 2: The architecture of our proposed neural joint model for Sindhi POS tagging and NER. The left part is Sindhi NER space, and the right part is Sindhi POS tagging private space, including embedding layer, private BiLSTM, self-attention, and CRF layer. The middle part is a shared space consisting of a shared BiLSTM, task discriminator, and self-attention.

3.4 CRF

The CRF learns the scoring function from tag pairs [42] as well as efficiently decodes the best chain of tags from a given input sequence. The self-attention mechanism is backbone model for joint tagging, used to encode vector h_i for each X_i. For h_i, we can employ many encoders, e.g., LSTM [25] or BERT [17] in order to get vector list $h_1 h_2, \cdots, h_i, \cdots, h_l]$ for X. For a given sentence in the dataset of task k, the final representation is computed by concatenating representations from both shared space and private space [12] after attention layer:

$$H''^k = H'^k \parallel Z'^k \tag{9}$$

where H'^k and Z'^k are the outputs of private self-attention and shared self-attention of each task k, respectively.

Due to the difference in tags, the modeling of independent assumptions would be impossible. Thus, we employ task-specific CRF for each task to inference tags instead of making tagging decisions using h'' independently to consider the correlation in the neighboring tags. For a given sentence $x = w_1 w_2, \cdot, w_2$, the probability of a predicted tag sequence $y = (y_1, y_2, \cdots, y_n)$ as follows:

$$o_i = W_z h''_i + b_z \tag{10}$$

$$z(x, y) = \sum_{i=1}^{N} \left(o_{i,y_i} + T_{y_{i-1}, y_i} \right) \tag{11}$$

$$\overline{y} = \arg\max_{y \in Y_x} z(x, y) \tag{12}$$

here W_z and b_z are trainable parameters. T is the transition matrix defines the scores of two successive tags z. $0_{i,y_i}$ denotes the score of y_i-th tag of the word w_i. Whereas Y_x denotes all candidate tag sequences for x. We use Viterbi algorithm for decoding to get predicted tag sequence $\overline{y}$.

We use the negative log-likelihood function for training and compute the probability of ground-truth tag sequence by:

$$p(\hat{y} \mid x) = \frac{e^{s(x,\hat{y})}}{\sum_{\widetilde{y} \in Y_x} e^{s(x,\widetilde{y})}} \tag{13}$$

where $\hat{y}$ represents the ground-truth tag sequence.

3.5 Task Discriminator

The adversarial training is incorporated into shared space to ensure that specific features of both tasks do not exist in shared space. A task discriminator [12] estimates which task the sentence comes from. Such process can be expressed as:

$$z'^k = \text{Maxpooling}\left(Z'^k\right) \tag{14}$$

$$D\left(z'^k; \theta_d\right) = \text{softmax}\left(W_d z'^k + b_d\right) \tag{15}$$

here θ_d denotes the parameters of task discriminator. W_d, b_d are the trainable parameters, and K represents number of tasks.

We use an adversarial loss [12] to prevent specific features of POS tagging from creeping into shared space. The gradient reversal layer [20] is added below the softmax to tackle the optimization problem. In the training of the proposed model, we minimize task discriminator errors. The gradients will become opposed sign to adversarially encourage the shared feature extractor to learn syntactic and task-shared word information through the gradient reversal

Table 1: Statistics of the SiPOS and SiNER datasets. The number of sentences *Sent* is split into training, development, and text sets.

Ds.	Task	Train Sent.	Dev. Sent.	Test Sent.
SiPOS	Sindhi POS tagging	5428	679	677
SiNER	Sindhi NER	5232	654	654

layer. After training, the shared task discriminator and feature extractor reach a point where the discriminator cannot differentiate the tasks according to the representations learned from the shared feature extractor.

4 EXPERIMENTS AND RESULTS

This section provides an overview of the experimental setting of baseline models and proposed joint neural model architecture. We use TensorFlow [1] to implement baselines and neural models on a single GTX 1080-TITAN GPU to conduct all the experiments.

4.1 Datasets

To evaluate our proposed model on Sindhi POS tagging and NER, we employ recently proposed two benchmark datasets (Ds.) of SiPOS [4] and SiNER [8] in the experiments. The SiPOS dataset is based on 16 universal POS tags consists of more than 2.93 million tokens, and has been manually annotated using the Doccano text annotation tool. The SiNER is also a gold-standard dataset for Sindhi NER, annotated on news corpus using the BIO tagging scheme. It contains more than 1.35 million tokens, but we use 0.293 million tokens to balance both datasets. The statistics of SiPOS and SiNER datasets is presented in Table 1, respectively. Both datasets are split into training, validation, and test sets.

4.2 Training and Evaluation

In training, similar hyper-parameters in all the experiments to analyze the performance difference. The hidden states of the BiLSTM encoder are set to 200, use the word-level and character-level pretrained subword embeddings [5] to initialize input representations. The maximum length of the input character sequence is set to 200 and use the negative log-likelihood loss function. We choose the dropout rate to 0.25%, and the initial learning rate is set to 0.005. We apply the dropout [50] of 0.25% to avoid the overfitting problem through all the experiments. Firstly, we select a task from at each iteration $\{POS, NER\}$, then sample a batch of training instances from the given task to update the parameters. We use Adamax algorithm [35] to optimize the final loss function. According to the Sindhi NER task performance, the iterations are repeated until early stopping because Sindhi NER and POS tagging may have different convergence rates. The rest of the baseline parameters and proposed models are tuned on the development set, and tuned models are evaluated on the test set. We use precision (P), recall (R), and F1-score metrics for the evaluation.

Table 2: Experimental results (F1-scores for POS and joint tagging) of baselines and joint neural model with BiLSTM encoder using different variants on benchmark datasets.

Model	POS	Joint
BiLSTM-CRF	93.76	88.43
BiLSTM-CRF-Transfer	94.57	89.86
BiLSTM-CRF-Adversarial	94.62	90.23
BiLSTM-CRF-Self-attention	95.82	89.75
BiLSTM-CRF-self-attention-Adversarial	**96.18**	**91.42**

4.3 Baselines

Table 2 presents the empirical results of our proposed joint neural model and baselines as simplified models on SiPOS and joint task, respectively. The baseline models are explained as follows:

- **BiLSTM-CRF:** The model is our initial and strong baseline that is trained using the Sindhi NER training set.
- **BiLSTM-CRF–Transfer:** We employ transfer learning to the initial baseline of the BiLSTM-CRF without using the self-attention and adversarial loss.
- **BiLSTM-CRF –Adversarial:** Compared with the BiLSTM-CRF-Transfer model, the BiLSTM-CRF-Adversarial combines adversarial training.
- **BiLSTM-CRF-Self-attention:** The model incorporates the self-attention based on the BiLSTM-CRF model.

4.4 Final Results

Table 2 shows the experimental results of our proposed model and baselines on POS tagging and joint task. We observe that our proposed model achieves new state-of-the-art performance on POS tagging and joint tasks. It is analyzed that the effectiveness of transfer learning in BiLSTM-CRF-transfer improves F1-scores from 93.76, 88.43 to 96.18, 91.42 as compared with BiLSTM-CRF for SiPOS dataset and joint task, respectively. The improvement indicates the contextual, syntactic, and word boundary information from the POS tagging is effective for the Sindhi NER task. Moreover, the adversarial training method is also an effective approach. The BiLSTM-CRF-Adversarial boosts the performance as compared with the BiLSTM-CRF-Transfer model, showing +0.37 improvement in the joint task. It proves that adversarial training can prevent specific features of POS tagging from creeping into shared space. Furthermore, the self-attention significantly increases the performance compared with the BiLSTM-CRF. The BiLSTM-CRF-Self-attention model yields better results on the two different datasets, which confirms that the self-attention is an effective approach for POS tagging as well as a joint task.

Moreover, we also compare our proposed model with the recent models on SiPOS and SiNER datasets. The comparative results in Table 3 and Table 4 demonstrate that our proposed joint model yields better performance than previous state-of-the-art Sindhi POS tagging and NER methods. The first block in Table 3 gives the performance of previous baselines and main models [4], the second block shows the performance of our proposed model. For the Sindhi NER task (see Table 4), the first block presents the performance of baselines [8], and the second block shows the results of baselines

Table 3: Sindhi POS tagging results on the original SiPOS dataset. The first block contain the baseline results [8] and the second block shows the results of our proposed joint model.

Model	P%	R%	F1%
CRF	90.43	91.28	91.36
LSTM	91.74	92.31	92.28
BiLSTM	92.19	92.88	92.76
BiLSTM-CRF	92.86	93.26	92.94
BiLSTM-CRF+Char	94.78	95.14	94.26
Ours	**96.36**	**95.82**	**96.18**

Table 4: Comparison of the Sindhi NER results (F1-scores) on the original SiNER dataset [8]. The first two blocks contain the baseline [8], simplified, and main models [7], respectively. The third block shows the results of our proposed joint model.

Model	P%	R%	F1%
CRF [8]	84.77	83.25	82.54
BiLSTM [8]	86.87	87.82	87.07
BiLSTM-CRF [8]	89.72	86.94	88.09
LSTM [7]	85.23	85.94	85.24
BiLSTM [7]	87.90	87.34	87.16
BiLSTM-CRF [7]	89.36	89.17	88.24
BiLSTM-CRF+Attn [7]	89.64	89.57	89.23
CaBiLSTM-CRF [7]	90.27	90.64	90.11
Ours	**91.64**	**91.83**	**91.42**

as well as main models [7], respectively. The third block presents the results of our proposed model on Sindhi NER. The overall improvement of (+1.92%) is achieved in F1-score as compare to previous Sindhi POS tagging [4] results and (+3.33%), (+1.31%) improvement as compare to previous state-of-the-art Sindhi NER [7, 8] methods. All Sindhi POS tagging models [4] are based on word-level representation learning, and all the NER models [7, 8] are based on character-level as well as word-level representation learning. We also use both word-level as well as character-level features in the pretrained embedding.

5 CONCLUSION AND FUTURE WORK

We propose a neural joint model for Sindhi POS tagging and NER, which takes advantage of the BiLSTM network, self-attention, and adversarial transfer learning. The key features of the proposed neural model are based on BiLSTM over the input word and character sequences. We observe that the self-attention mechanism and adversarial transfer learning lead to noticeable improvements over the baseline. Self-attention captures the dependencies of arbitrary two tokens and learns the inner structure of the sentence. Thus, the proposed model efficiently exploit task-shared features between two different tasks. Empirical results demonstrate that our proposed method consistently and significantly outperforms the existing models. We intend to train Sindhi deep contextualized representation models of Elmo, BERT, and GPT for multitasking in the future.

ACKNOWLEDGMENTS

We thank to the anonymous reviewers for concrete suggestions. This work was funded by the National Key R&D Program of China (No. 2018YFB1005100 & No. 2018YFB1005104).

REFERENCES

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. Tensorflow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI) 16)*. 265–283.

[2] Wazir Ali, Asadullah Kehar, and Hidayatullah Shaikh. 2015. Towards Sindhi Named Entity Recognition: Challenges and opportunities. In *1st National Conference on Trends and Innovations in Information Technology*.

[3] Wazir Ali, Asadullah Kehar, and Hidayatullah Shaikh. 2016. Towards Sindhi named entity recognition: Challenges and opportunities. (2016).

[4] Wazir Ali, Jay Kumar, , Rajesh Kumar, and Zenglin Xu. 2021. SiPOS: A Benchmark Part-of-Speech Tagset for Low-resourced Language: Sindhi. (2021).

[5] Wazir Ali, Jay Kumar, Junyu Lu, and Zenglin Xu. 2020. Word Embedding based New Corpus for Low-resourced Language: Sindhi. *arXiv preprint arXiv:1911.12579* (2020).

[6] Wazir Ali, Jay Kumar, Zenglin Xu, Congjian Luo, Junyu Lu, Junming Shao, Rajesh Kumar, and Yazhou Ren. 2020. A Subword Guided Neural Word Segmentation Model for Sindhi. arXiv:2012.15079 [cs.CL]

[7] Wazir Ali, Jay Kumar, Zenglin Xu, Junming Shao, and Yazhou Ren. 2021. A Deep Context-aware Bidirectional Neural Model for Sindhi Named Entity Recognition. (2021).

[8] Wazir Ali, Junyu Lu, and Zenglin Xu. 2020. SiNER: A Large Dataset for Sindhi Named Entity Recognition. In *Proceedings of The 12th Language Resources and Evaluation Conference*. European Language Resources Association, 2946–2954.

[9] Daniel M Bikel, Scott Miller, Richard Schwartz, and Ralph Weischedel. 1997. Nymble: a High-Performance Learning Name-finder. In *Fifth Conference on Applied Natural Language Processing*. 194–201.

[10] Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. 2016. Domain Separation Networks. In *Proceedings of the Neural Information Processing Systems*.

[11] Tihana Britvić. 2018. *Semi-supervised neural part-of-speech tagging*. Ph.D. Dissertation. University of Zagreb. Faculty of Science. Department of Mathematics.

[12] Pengfei Cao, Yubo Chen, Kang Liu, Jun Zhao, and Shengping Liu. 2018. Adversarial transfer learning for Chinese named entity recognition with self-attention mechanism. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 182–192.

[13] Xinchi Chen, Zhan Shi, Xipeng Qiu, and Xuan-Jing Huang. 2017. Adversarial Multi-Criteria Learning for Chinese Word Segmentation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 1193–1203.

[14] Michael Collins. 2002. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the ACL-02 conference on Empirical Methods in Natural Language Processing*, Vol. 10. 1–8.

[15] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural Language Processing (Almost) from Scratch. *The Journal of Machine Learning Research* 12 (2011), 2493–2537.

[16] Rômulo César Costa de Sousa and Hélio Lopes. 2019. Portuguese POS Tagging Using BLSTM Without Handcrafted Features. In *Iberoamerican Congress on Pattern Recognition*. Springer, 120–130.

[17] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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*. 4171–4186.

[18] Mazhar Ali Dootio and Asim Imdad Wagan. 2019. Syntactic parsing and supervised analysis of Sindhi text. *Journal of King Saud University-Computer and Information Sciences* 31, 1 (2019), 105–112.

[19] Jeffrey L Elman. 1990. Finding structure in time. *Cognitive science* 14, 2 (1990), 179–211.

[20] Yaroslav Ganin and Victor Lempitsky. 2015. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*. PMLR, 1180–1189.

[21] Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning Word Vectors for 157 Languages. In *Language Resources and Evaluation Conference*.

[22] Tao Gui, Qi Zhang, Haoran Huang, Minlong Peng, and Xuan-Jing Huang. 2017. Part-of-speech tagging for twitter with adversarial neural networks. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 2411–2420.

[23] Onur Güngör, Suzan Uskudarli, and Tunga Güngör. 2018. Improving Named Entity Recognition by Jointly Learning to Disambiguate Morphological Tags. In *Proceedings of the 27th International Conference on Computational Linguistics*. 2082–2092.

[24] Maqsood Ahmed Hakro, Intzar Ali Lashari, et al. 2017. Sindhi Named Entity Recognition (SNER). *The Government-Annual Research Journal of Political Science*. 5, 5 (2017).

[25] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[26] Chaoyi Huang, Youguang Chen, and Qiancheng Liang. 2019. Attention-based bidirectional long short-term memory networks for Chinese named entity recognition. In *In Proceedings of the 4th International Conference on Machine Learning Technologies*. 53–57.

[27] Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF Models for Sequence Tagging. *arXiv preprint arXiv:1508.01991* (2015).

[28] Hideki Isozaki and Hideto Kazawa. 2002. Efficient support vector classifiers for named entity recognition. In *COLING 2002: The 19th International Conference on Computational Linguistics*.

[29] Wazir Ali Jamro. 2017. Sindhi Language Processing: A survey. In *International Conference on Innovations in Electrical Engineering and Computational Technologies*. 1–8.

[30] Yaozong Jia and Xiaopan Ma. 2019. Attention in character-Based BiLSTM-CRF for Chinese named entity recognition. In *In Proceedings of the 4th International Conference on Mathematics and Artificial Intelligence*. 1–4.

[31] Awais Khan Jumani, Mashooque Ahmed Memon, Fida Hussain Khoso, Anwar Ali Sanjrani, and Safeeullah Soomro. 2018. Named entity recognition system for Sindhi language. In *International conference for emerging technologies in computing*. Springer, 237–246.

[32] Katharina Kann, Johannes Bjerva, Isabelle Augenstein, Barbara Plank, and Anders Søgaard. 2018. Character-level supervision for low-resource POS tagging. In *Proceedings of the Workshop on Deep Learning Approaches for Low-Resource Natural Language Processing*. 1–11.

[33] Wahab Khan, Ali Daud, Jamal Abdul Nasir, Tehmina Amjad, Sachi Arafat, Naif Aljohani, and Fahd S Alotaibi. 2019. Urdu part of speech tagging using conditional random fields. *Language Resources and Evaluation* 53, 3 (2019), 331–362.

[34] Joo-Kyung Kim, Young-Bum Kim, Ruhi Sarikaya, and Eric Fosler-Lussier. 2017. Cross-lingual transfer learning for pos tagging without cross-lingual resources. In *Proceedings of the 2017 conference on empirical methods in natural language processing*. 2832–2838.

[35] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[36] Jay Kumar, Rajesh Kumar, Amin Ul Haq, and Sidra Shafiq. 2020. A Non-Parametric Multi-Lingual Clustering Model for Temporal Short Text. In *2020 17th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*. IEEE, 58–61.

[37] John D Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the Eighteenth International Conference on Machine Learning*. 282–289.

[38] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 260–270.

[39] Hao Liu, Lirong He, Haoli Bai, Bo Dai, Kun Bai, and Zenglin Xu. 2018. Structured Inference for Recurrent Hidden Semi-markov Model. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*. 2447–2453.

[40] Maofu Liu, Yukun Zhang, Wenjie Li, and Donghong Ji. 2020. Joint Model of Entity Recognition and Relation Extraction with Self-attention Mechanism. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)* 19, 4 (2020), 1–19.

[41] Pengfei Liu, Xipeng Qiu, and Xuan-Jing Huang. 2017. Adversarial Multi-task Learning for Text Classification. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 1–10.

[42] Xuezhe Ma and Eduard Hovy. 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*. 1064–1074.

[43] Javed Ahmed Mahar and Ghulam Qadir Memon. 2010. Rule based part of speech tagging of sindhi language. In *2010 International Conference on Signal Acquisition and Processing*. IEEE, 101–106.

[44] Javed Ahmed Mahar and Ghulam Qadir Memon. 2010. Sindhi part of speech tagging system using wordnet. *International Journal of Computer Theory and Engineering* 2, 4 (2010), 538.

[45] Raveesh Motlani, Harsh Lalwani, Manish Shrivastava, and Dipti Misra Sharma. 2015. Developing part-of-speech tagger for a resource poor language: Sindhi. In *Proceedings of 7th Conference on Language and Technology, Poznan, Poland*.

[46] D Nawaz, SA Awan, ZA Bhutto, M Memon, and M Hameed. 2017. Handling Ambiguities in Sindhi Named Entity Recognition (SNER). *Sindh University Research Journal-SURJ (Science Series)* 49, 3 (2017), 513–516.

[47] Cicero D Santos and Bianca Zadrozny. 2014. Learning character-level representations for part-of-speech tagging. In *Proceedings of the 31st international conference on machine learning (ICML-14)*. 1818–1826.

[48] Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing* 45, 11 (1997), 2673–2681.

[49] Irum Naz Sodhar, Akhtar Hussain Jalbani, Muhammad Ibrahim Channa, and Dil Nawaz Hakro. 2019. Parts of Speech Tagging of Romanized Sindhi Text by applying Rule Based Model. *International Journal of Computer Science and Network Security* 19, 11 (2019), 91.

[50] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research* 15, 1 (2014), 1929–1958.

[51] Farhan Ali Surahio and Javed Ahmed Mahar. 2018. Prediction system for sindhi parts of speech tags by using support vector machine. In *2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*. IEEE, 1–6.

[52] Masaya Suzuki, Kanako Komiya, Minoru Sasaki, and Hiroyuki Shinnou. 2018. Fine-tuning for Named Entity Recognition Using Part-of-Speech Tagging. In *Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation*.

[53] Ioannis Tsochantaridis, Thomas Hofmann, Thorsten Joachims, and Yasemin Altun. 2004. Support vector machine learning for interdependent and structured output spaces. In *Proceedings of the twenty-first international conference on Machine learning*. 104.

[54] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*. 5998–6008.

[55] YaoSheng Yang, Meishan Zhang, Wenliang Chen, Wei Zhang, Haofen Wang, and Min Zhang. 2018. Adversarial learning for Chinese NER from crowd annotations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.

[56] Meishan Zhang, Nan Yu, and Guohong Fu. 2018. A simple and effective neural model for joint word segmentation and POS tagging. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 26, 9 (2018), 1528–1538.

[57] Yuan Zhang, Regina Barzilay, and Tommi Jaakkola. 2017. Aspect-augmented adversarial networks for domain adaptation. *Transactions of the Association for Computational Linguistics* 5 (2017), 515–528.

[58] Junsheng Zhou, Weiguang Qu, and Fen Zhang. 2013. Chinese named entity recognition via joint identification and categorization. *Chinese journal of electronics* 22, 2 (2013), 225–230.