International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

POS Tagging of Sindhi Language Using Subword Representations and Neural Models

Jagroop Kaur[1], Himani Gupta[2], Gurpreet Singh Josan[3]

Submitted: 01/01/2023 **Revised:** 05/02/2023 **Accepted:** 15/02/2023

Abstract: Part-of-Speech (POS) tagging is a fundamental Natural Language Processing (NLP) task that facilitates a wide range of downstream applications. While extensive research exists for resource-rich languages such as English and Chinese, morphologically rich and low-resource languages like Sindhi remain underexplored. This paper presents POS tagging models for Sindhi using word-level, character-level, joint word-character, and subword-level representations. To address challenges such as ambiguity, semantic preservation, and out-of-vocabulary (OOV) words, we employ Byte Pair Encoding (BPE) based subword representations in combination with Bidirectional Long Short-Term Memory (BiLSTM) networks. Two classifier settings are evaluated: a Dense layer and a Conditional Random Field (CRF) layer. Experiments are conducted on publicly available Sindhi datasets (SiPOS and Dootio-Wagan), with Dataset-1 used for training and Dataset-2 for evaluation. Results show that joint word-character BiLSTM-CRF achieves the highest accuracy (90%), while the proposed BPE-based subword BiLSTM-Dense model achieves 88%, outperforming the subword BiLSTM-CRF at 86%. These findings demonstrate that subword representations effectively handle OOV and morphological complexity while retaining semantic information. The proposed models enrich Sindhi computational resources and highlight promising directions for future work, including training Sindhi-specific BPE embeddings and exploring transformer-based architectures such as RoBERTa and GPT-2 for improved accuracy.

Keywords: *Part of speech, tagging, word representations, subword representations, Byte Pair Encoding, BiLSTM, neural network*

1. Introduction

One of the extensively studied problems in the field of Natural Language Processing (NLP) is sequence-to-sequence tagging. Part-of-speech (POS) tagging, a fundamental sequence labeling task, involves classifying words according to their grammatical categories and assigning corresponding labels from a predefined tagset. The characteristics of a word that define its role, meaning, and usage within a sentence are referred to as its parts of speech. POS tagging operates at the token level. Sindhi grammar has long been standardized, and linguists have identified eight major parts of speech: noun, pronoun, verb, adjective, adverb, preposition, conjunction, and interjection [Mahar and Memon, 2010].

Automating POS tagging has been a persistent challenge in the NLP community. Manual annotation is not only time-consuming but also prone to human error, making automation essential. However, POS tagging faces multiple challenges. The first is ambiguity, where words behave differently depending on the context, making it difficult to determine the correct tag [Cliche and Yitagesu, 2022]. For instance, the Sindhi word خواهه (khwahh, "want") may be tagged as an adjective, proper noun, or pronoun, depending on usage. Another challenge is accurate semantic extraction, which is critical for understanding word relationships and capturing context-sensitive features.

A further difficulty arises from out-of-vocabulary (OOV) words, which are absent from training data but appear during testing. Such words are often assigned zero probability by models, significantly degrading performance. To address this, our work employs the Byte Pair Encoding (BPE) algorithm [Gage, 1994], which segments words—including OOV terms—into subwords, thereby reconstructing their meanings from constituent parts. This effectively mitigates the OOV problem.

Sindhi, a morphologically rich language, dates back to the 8th century AD in written form. Historically, it has been written in multiple scripts: Landa-derived scripts such as Khojki and Khudabadi, as well as Gurmukhi, Devanagari, Perso-Arabic, and Roman. Different communities adopted different scripts—for example, Pandits used Devanagari, Hindu women employed Gurmukhi, and by the 19th century, Perso-Arabic became the official script. Today, Perso-Arabic, Devanagari, and Roman remain widely used.

In this paper, we propose a POS tagging model for Sindhi that leverages subword representations generated using BPE [Gage, 1994]. BPE is a data compression technique that iteratively replaces the most frequent pair of consecutive bytes with a single unused byte. Applied to text, this reduces vocabulary size by merging frequent character sequences into tokens, while rare words are decomposed into smaller subwords. Unlike traditional

1 Department of Computer Science and Eng, Punjabi University, Patiala
ORCID ID : 0000-0002-5572-1919
2 Department of Computer Science, Punjabi University, Patiala
ORCID ID : 0009-0007-7641-2289
3 Department of Computer Science, Punjabi University, Patiala
ORCID ID : 0000-0002-3195-2229
** Corresponding Author Email: josanjagroop80@gmail.com*

word-level tokenization, which requires a large vocabulary and struggles with OOV words, or character-level tokenization, which risks losing semantic information, subword-based methods balance both aspects by retaining semantic features while reducing vocabulary size.

We employ BPEmb [Heinzerling and Strube, 2018], a collection of pre-trained subword embeddings for 275 languages—including Sindhi—trained on Wikipedia using BPE. These embeddings serve as input to a deep learning architecture comprising a Bidirectional Long Short-Term Memory (BiLSTM) network [Schuster and Paliwal, 1997] for feature extraction, followed by a dense layer and a Conditional Random Field (CRF) classifier.

This paper explores how BPE-based subword representations improve Sindhi POS tagging by addressing the key challenges of ambiguity, semantics, and OOV words. Section 2 reviews related work, Section 3 describes the dataset and preprocessing steps, Section 4 details the experimental setup, Section 5 presents results, and Section 6 concludes with key findings.

2. Related Work

Early work on part-of-speech (POS) tagging was dominated by rule-based approaches. Brill [1992] introduced a transformation-based tagger that automatically acquired tagging rules and achieved accuracy comparable to stochastic taggers. As probabilistic methods gained prominence, Conditional Random Fields (CRFs) emerged as a powerful alternative. Lafferty et al. [2001] first applied CRFs to POS tagging, demonstrating their superiority over Maximum Entropy Markov Models (MEMMs). CRFs were subsequently applied to shallow parsing by Sha and Pereira [2003], who showed their effectiveness in noun phrase chunking on the Wall Street Journal corpus. For morphologically rich languages such as Gujarati, Patel and Gali [2008] used CRFs to integrate diverse linguistic features into tagging models.

For Sindhi, initial efforts were largely rule-based. Mahar and Memon [2010] developed a POS tagger by designing a tagset, lexicon, and word disambiguation rules, along with tokenization algorithms verified by Sindhi linguists. Later advances followed broader trends in neural approaches to sequence labeling. Santos and Zadrozny [2014] proposed a deep neural network that combined character-level representations with word embeddings for POS tagging, an approach that addressed out-of-vocabulary (OOV) issues. In parallel, Sindhi resources began to expand. Motlani et al. [2015] introduced a Sindhi corpus in Devanagari script with 44,000 tokens, while Dootio and Wagan [2018] contributed a dataset of 6,841 lexical entries for computational applications.

Recognizing the low-resource status of Sindhi, Ali et al. [2021] released the SiPOS benchmark dataset, consisting of over 293,000 tokens annotated with 16 Universal POS (UPOS) categories. Their evaluation incorporated CRFs, BiLSTMs, and self-attention models, leveraging pre-trained embeddings such as GloVe and fastText along with character-level features. Other relevant efforts include Sodhar et al. [2021], who formalized rules for Romanized Sindhi text communication, and Warjari et al. [2021], who developed a Khasi corpus and POS taggers using BiLSTM, BiLSTM–CRF, and character-based embeddings.

Neural architectures have become central to POS tagging. Schuster and Paliwal [1997] introduced BiLSTMs, which capture both prefix and suffix contexts by processing input bidirectionally. When combined with CRF classifiers, BiLSTM–CRF architectures achieved state-of-the-art performance in sequence labeling tasks [Huang et al., 2015]. Character-level models further enhanced tagging accuracy, with Dos Santos and Zadrozny [2014] showing significant improvements by encoding morphological features. More recently, subword modeling has gained traction, particularly for morphologically rich and low-resource languages. Byte Pair Encoding (BPE) [Gage, 1994], adapted for tokenization, reduces vocabulary size and effectively handles OOV words. Heinzerling and Strube [2018] extended this to BPEmb, a library of pre-trained BPE embeddings for over 270 languages, including Sindhi.

In summary, POS tagging research has evolved from rule-based and stochastic methods to neural architectures and subword approaches. For Sindhi, however, progress has been constrained by limited annotated corpora and script diversity (Perso-Arabic, Devanagari, Romanized). While recent resources such as SiPOS and BPEmb have begun to fill this gap, POS tagging for Sindhi remains relatively underexplored. This highlights the need for further development of robust datasets and advanced tagging models to fully address the challenges posed by Sindhi's morphological richness and low-resource status.

3. Data And Preprocessing

3.1. Data Set

For this research, two publicly available datasets for Sindhi were utilized. The first dataset, SiPOS, was introduced by Ali et al. [2021]. It consists of over 293,000 tokens annotated with both Universal POS (UPOS) and Sindhi POS (SPOS) attributes, each covering 16 tag categories. This dataset is referred to as Dataset-1. The second dataset was developed by Dootio and Wagan [2018]. It is an UTF-8 encoded CSV corpus containing 6,841 records with 19 attributes, including UPOS, SPOS, EqlNumUPOS, EqlNumSPOS, gender (1M/2F), number (1S/2P), polarity (1p/2n/3nu), sentiment labels (Positive, Negative, Neutral), morphological features (Morpho1P/2S/0N, complex word, compound word, reduplicated word), lemma, diacritics, infinitive form, unigram probability, and token. This dataset includes 18 UPOS tag categories and is referred to as Dataset-2. In this study, Dataset-1 is used for training, while Dataset-2 serves as the evaluation set. A summary of both datasets is presented in Table 1.

Table 1. Dataset Overview

Data property	Dataset-1	Dataset-2
No. of Tokens	293680	6841
No. of sentences	6397	692
Max. sentence length	327	69
Average sentence length	45(approx.)	9(approx.)

3.2. Pre-Processing

Dataset-1 contains some unreadable characters, which were removed during the data cleaning process. This dataset is annotated with 16 POS tags. In contrast, Dataset-2 follows the UPOS tagset with 18 tags, while Dataset-1 uses a different tagging scheme with only 16 categories. Therefore, a mapping step is required to align the 18 UPOS tags of Dataset-2 with the 16 tags of Dataset-1. The tag mappings are presented in Table 2.

Table 2.Tags of Dataset-2 mapped to tags in Dataset-1

Sr.No.	Tag in Dataset-2	Mapped to tag in Dataset-1
1.	PUNC	PUNCT
2.	PERIOD	PUNCT

3.	SYM	PUNCT
4.	X	UNK
5.	-	FOW
6.	VERB	VB
7.	NOUN	NN
8.	PROPN	NNP

After pre-processing, Dataset-2 contains 15 tags. The tag count mapping between Dataset-1 and Dataset-2 is shown in Table 3. During pre-processing, all extra leading and trailing spaces were removed. Furthermore, the data was transformed from a column-wise word format into lists of sentences using end-of-sentence markers such as ".", "।", "؟", "!", or "۔". For example, a pre-processed sentence appears as: [' زندگي', 'گذاريندو', 'نه', 'آهيان', 'بلڪه', 'ان', 'کي', 'جيئندو', 'آهيان', '.'] The corresponding POS tags for this sentence after pre-processing are: ['NN', 'VB', 'ADV', 'AUX', 'CONJ', 'DET', 'ADP', 'VB', 'AUX', 'PUNCT']

Table 3. Tag count in Dataset-1 and Dataset-2

TAG	POS type	Count (Dataset-1)	Count (Dataset-2)
NN	Noun	65611	1566
ADP	Preposition	54810	1042
VB	Verb	41882	1069
ADJ	Adjective	21849	594
DET	Determiner	20627	231
PUNCT	Punctuation	20277	732
ADV	Adverb	19823	334
NNP	Proper Noun	11546	308
CONJ	Conjunction	11015	167
AUX	Auxillary Verb	10553	405
PRON	Pronoun	7878	274
NUM	Numerical adjective	4888	44
FOW	Borrowed words	1424	-
UNK	Unknown	1091	63
SCON	Subordinating conjunction	282	5
INTJ	Interjection	124	8

4. Experimental Setup

4.1 Types of Representations Used

Word representations: Vector-based representations of words that aim to capture their semantic meaning.

Character representations: Character-level representations can effectively handle out-of-vocabulary (OOV) words, as they allow inference of unseen words. Due to the smaller vocabulary size, models trained on character representations are also computationally efficient.

Subword representations: Subword-level approaches mitigate the unknown word problem by constructing word meaning from smaller units. They retain semantic features, handle OOV words, and allow tokenization without requiring an excessively large vocabulary.

4.2 Model Development

4.2.1. Word Model

Embedding: Pre-processed sentences are converted into word sequences and passed through an embedding layer, which maps each word to a fixed-length vector.

Feature Extraction: The embedded vectors are passed through two layers of BiLSTM to extract sequential features.

Classifier: The resulting features are classified using two alternative classifiers: (i) a Dense layer, and (ii) a CRF layer.

4.2.2 . Character Model

This model follows the same architecture as the word model, except that character sequences are used instead of word sequences. Features are extracted by passing the character embeddings through BiLSTM layers.

4.2.3. Word-character joint Model

Embedding: Both word and character sequences are embedded separately.

Feature Extraction: Word-level and character-level embeddings are concatenated and passed through two BiLSTM layers to learn joint features.

Classifier: The extracted features are passed to a classifier to predict the POS tags. Two classifiers are evaluated: (i) a Dense layer classifier, and (ii) a CRF classifier.

4.2.4 Subword Model

Embedding: Pre-processed sentences are converted into subword sequences using the BPEmb module for Sindhi [Heinzerling and Strube, 2018]. These sequences are passed through an embedding layer initialized with pre-trained BPEmb Sindhi vectors.

Feature Extraction: The subword embeddings are fed into two stacked BiLSTM layers to extract sequential features.

Classifier: The extracted features are passed to the classifier for tag prediction. As with other models, two classifiers are tested: a Dense layer classifier and a CRF classifier.

This work introduces a **subword-based POS tagging model** for Sindhi. The architecture of the proposed subword model is illustrated in **Figure 1**.

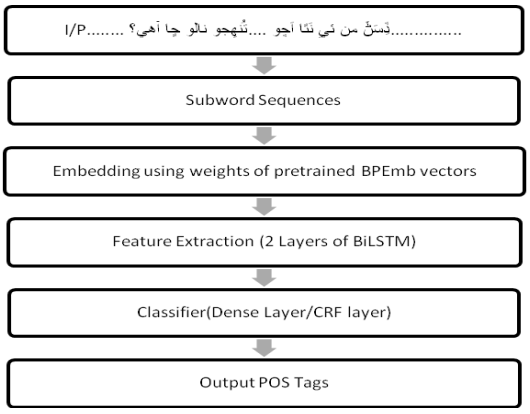

Fig.1. The Model Architecture

The deep neural network architecture employed in this research for Sindhi POS tagging is the Bidirectional Long Short-Term Memory (BiLSTM) network. Two variants of classifiers are explored: a Dense layer and a Conditional Random Field (CRF) layer. BiLSTM, first introduced by Schuster and Paliwal [1997], has been widely adopted for sequence labeling tasks. Unlike traditional LSTMs, BiLSTM processes input in both directions—left-to-right and right-to-left. When applied to text, this allows the model to learn from both the prefix (left context) and the suffix (right context) of a target word, thereby producing a richer representation for final classification. The Dense layer is a fully

connected neural network layer in which each neuron receives input from all neurons in the preceding layer. It performs a matrix–vector multiplication followed by a non-linear transformation. In this work, features extracted from two BiLSTM layers are passed to the Dense layer to obtain the predicted POS tags. The Conditional Random Field (CRF) is a discriminative sequence modeling technique that establishes decision boundaries between classes while leveraging contextual information from neighboring labels. By considering dependencies across output sequences, the CRF helps the model make more consistent and accurate predictions. In the BiLSTM–CRF architecture, features extracted from two BiLSTM layers are fed into the CRF layer, which outputs the final sequence of POS tags for Sindhi text.

4.3 Training & testing

Dataset-1 is used as training data and Dataset-2 is used for testing and evaluation purpose. Evaluation metrics like Precision, Recall, F1 score and accuracy are used for evaluation.

4.3.1. Hyper-parameters

Several hyperparameter configurations were explored during evaluation, and the optimal settings were selected based on the highest achieved accuracy. For the BiLSTM–Dense subword model, the Adam optimizer [Kingma and Ba, 2015] was used with a learning rate of 0.01. The embedding layer was configured with an input dimension of 10,000 and an output dimension of 300. The model employed two BiLSTM layers, each with 256 hidden units, followed by a dropout layer with a rate of 0.2. The Dense layer used softmax activation for classification. The categorical cross-entropy loss function was applied. Training was performed with a batch size of 256 for up to 30 epochs, using early stopping based on validation loss and a checkpoint monitor on validation accuracy. For the BiLSTM–CRF subword model, the Adam optimizer was used with a lower learning rate of 0.001. The embedding layer was configured with the same input (10,000) and output (300) dimensions as in the BiLSTM–Dense model. Two BiLSTM layers with 256 hidden units each were employed, and the features extracted from these layers were passed to a CRF layer consisting of 16 units, corresponding to the 16 POS tags in the training dataset. The model was trained with the Sigmoid Focal Cross-Entropy loss function, a batch size of 256, and up to 45 epochs, with early stopping and checkpoint monitoring based on validation accuracy.

5. Results And Analysis

This section presents the results of the BiLSTM–Dense and BiLSTM–CRF models for POS tagging of Sindhi, evaluated using different input representations: word-based, character-based, joint word–character, and subword representations. The performance outcomes of these models are summarized in Table 4, which reports the results for all four representation types. All models were trained on Dataset-1 and evaluated on Dataset-2.

Table 4. Results of BiLSTM-Dense model and BiLSTM-CRF

Representation type	Models	Test data: Dataset-2			
		Prec	Recall	F1 score	Acc.
Word representation	BiLSTM-Dense	0.89	0.85	0.87	89%
	BiLSTM-CRF	0.89	0.82	0.82	89%
Character representation	BiLSTM-Dense	0.86	0.79	0.81	90%
	BiLSTM-CRF	0.90	0.86	0.87	90%
Joint Word-Character representation	BiLSTM-Dense	0.90	0.87	0.88	89%
	BiLSTM-CRF	0.90	0.87	0.88	90%
Subword (BPE) representation	BiLSTM-Dense	0.85	0.77	0.79	88%
	BiLSTM-CRF	0.87	0.78	0.80	86%

The word representation models with both Dense and CRF classifiers achieve an accuracy of 89%. The character representation models attain a slightly higher accuracy of 90% with both classifiers. For the joint word–character representation, the model with the CRF classifier achieves 90%, while the model with the Dense classifier reaches 89%. The subword representation models based on BPE show lower performance: the Dense classifier achieves 88%, which is still better than the CRF classifier at 86%. Overall, the character-based model and the joint word–character model with the CRF classifier perform the best, both reaching 90% accuracy. Among these, the joint word–character model with the CRF classifier can be considered the most effective, as it leverages both word- and character-level features. Figure 2 presents the confusion matrix showing precision per class for the joint word–character model with the CRF classifier, while Figure 3 illustrates the corresponding recall per class.

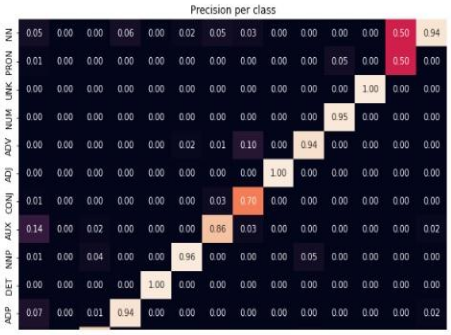

Fig.2. Confusion Matrix showing precision per class (Tag) for joint word-character model with CRF classifier.

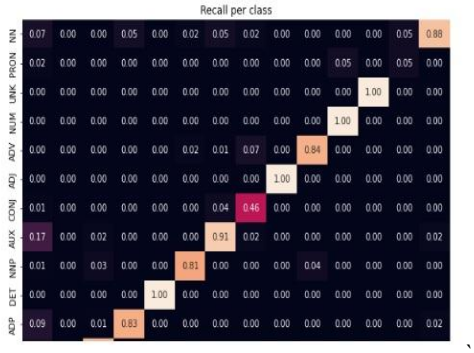

Fig.3. Confusion Matrix showing Recall per class (Tag) for joint Word-Character model with CRF classifier

5.1.1. Discussion

There exists an inconsistency between the tags of the training and

testing datasets, as they originate from different sources. For instance, the testing dataset does not contain any words corresponding to the tag "FOW".

The word representation model with the Dense classifier fails to identify words with the tags FOW, INTJ, SCON, and UNK. The word representation model with the CRF classifier is unable to recognize FOW and INTJ. The character representation models, which achieved the maximum accuracy of 90%, also face limitations: the Dense classifier version cannot predict FOW and INTJ, while the CRF classifier version misses FOW, INTJ, and UNK. Similarly, the joint word–character representation models with both Dense and CRF classifiers fail to predict FOW, INTJ, and UNK. For the subword representation models, the Dense classifier does not predict FOW and INTJ, while the CRF classifier fails on FOW, INTJ, and UNK.

The proposed subword-based POS tagging models achieve accuracies of 88% (Dense) and 86% (CRF). A key challenge lies in the way tags expand under the Byte Pair Encoding (BPE) algorithm. For each predicted tag, 16 values are generated, but only one is required. To address this, the tag of either the first or last token in a subword unit must be chosen as the final tag. In this work, the first token's tag was selected during evaluation, which could be a limiting factor. If a more refined method is employed—for example, selecting the most probable tag across subword units—the performance of subword-based models could improve substantially. Such improvements would allow these models to better handle challenges related to semantics, out-of-vocabulary words, and large vocabulary sizes.

6. Conclusion & Future Scope

This paper presents POS tagging models for the Sindhi language based on subword representations derived using Byte Pair Encoding (BPE). In addition, it evaluates models developed with three alternative approaches: word representations, character representations, and joint word–character representations. While these three approaches have been explored earlier by researchers, our work extends them with subword-based models. All models were implemented using BiLSTM neural networks, with either a Dense layer or a CRF layer as the classifier.

Among the models, the joint word–character BiLSTM-CRF model achieves the highest accuracy of 90%, making it the best-performing approach. The BiLSTM-Dense subword model achieves 88% accuracy, outperforming its BiLSTM-CRF counterpart and demonstrating the promise of subword-level representations for Sindhi POS tagging.

The proposed models represent a valuable addition to Sindhi language resources and the field of computational linguistics for Sindhi. For future work, the BPE algorithm could be pre-trained from scratch specifically for Sindhi to further enhance tagging performance. Moreover, leveraging pre-trained transformer models such as RoBERTa or GPT-2 could substantially improve accuracy. As more high-quality, user-friendly linguistic resources for Sindhi become available, we expect significant progress in POS tagging and broader NLP applications, particularly given the language's complex and morphologically rich nature. Addressing inconsistencies and discrepancies in resource creation will be crucial for advancing Sindhi NLP.

Author contributions

Jagroop Kaur: Methodology, Validation, corresponding author **Himani Gupta:** Data curation, Writing- draft preparation, Software, Experimentation **Gurpreet Singh Josan:** Conceptualization, Algorithm Design.

Conflicts of interest

The authors declare no conflicts of interest.

References

[1] Ali, w., xu, z., and kumar, j. 2021. Sipos: a benchmark dataset for sindhi part-of- speech tagging. *In proceedings of the student research workshop associated with ranlp 2021* , 22-30.

[2] Brill, e. 1992. A simple rule-based part of speech tagger . Proceedings of the third conference on applied computational linguistics (acl), trento .

[3] Cliche, a., and yitagesu, b. 2022. Part of speech tagging: a systematic review of machine learning and deep learning approaches. *Journal of big data , 9* (1).

[4] D.nawaz, awan, s. A., bhotto, z. A., memon, m., and hameed, m. 2017. Handling ambiguities in sindhi named entity recognition(ner). *Sindhi university research journal(science series) , 49* (3), 513-516.

[5] Dootio, m. A., and wagan, a. I. 2018. Unicode-8 based linguistics data set of annotated sindhi text. *Data in brief , 19*, 1504-1514.

[6] Gage, p. 1994. A new algorithm for data compression. *C user journal , 12* (2), 23-28.

[7] Heinzerling, b., and strube, m. 2018. Bpemb: tokenization-free pre-trained subword embeddings in 275 languages. *Proceedings of the eleventh international conference on language resources and evaluation ({lrec} 2018).* Miyazaki, japan: european language resources association (elra).

[8] Huang, z., xu, w., & yu, k. (2015). Bidirectional lstm-crf models for sequence tagging. Proceedings of the 2015 conference on empirical methods in natural language processing (emnlp), pp. 2261–2270.

[9] Kingma, d. P., and ba, j. 2015. Adam: a method for stochastic optimization. *3rd international conference for learning representations.* San diego: arxiv.

[10] Lafferty, j., mccallum, a., and pereira, f. 2001. Conditional random fields: probabilistic models for segmenting and labeling sequence data. *Proceedings of the eighteenth international conference on machine learning*, (pp. 282-289).

[11] Mahar, j. A., and memon, g. Q. 2010. Sindhi part of speech tagging system using wordnet. *International journal of computer theory and engineering , 2* (4), 1793-8201.

[12] Motlani, r., lalwani, h., sharma, d. M., and shrivastava, m. 2015. Developing part-of-speech tagger for a resource poor language: sindhi. *In proceedings of 7th conference on language and technology,ponzan,poland* .

[13] Patel, c., and gali, k. 2008. Part-of-speech tagging for gujarati using conditional random fields.

In *proceedings of the ijcnlp-08 workshop on nlp for less privileged languages*.

[14] Santos, c. D., and zadrozny, b. 2014. Learning character-level representations for part-of-speech tagging. *Proceedings of the 31st international conference on machine learning , 32* (2), 1818-1826.

[15] Schuster, m., and paliwal, k. K. 1997. Bidirectional recurrent neural networks. *Ieee trans.signal process. , 45*, 2673-2681.

[16] Sha, f., and pereira, f. 2003. Shallow parsing with conditional random fields. Proceedings of the 2003 human language technology conference of the north american chapter of the association for computational linguistics, (pp. 213-220).

[17] Sodhar, i., jalbani, a., channa, m., and hakro, d. 2021. Romanized sindhi rules for text communication. *Mehran university research journal of engineering and technology, 40*(2), 298 - 304. Doi:10.22581/muet1982.2102.04

[18] Warjari, s., pakray, p., lingdoh, s. A., and maji, a. K. 2021. Part-of-speech (pos) tagging using deep learning-based approaches on the designed khasi pos corpus. *Acm trans. Asian low-resour. Lang. Inf. Process. , 21* (3), 2375-4699.