Journal of King Saud University – Computer and Information Sciences 35 (2023) 335–356

HOSTED BY

Contents lists available at ScienceDirect

Journal of King Saud University – Computer and Information Sciences

journal homepage: www.sciencedirect.com

Neural POS tagging of shahmukhi by using contextualized word representations

Amina Tehseen [a], Toqeer Ehsan [b], Hannan Bin Liaqat [c], Amjad Ali [d], Ala Al-Fuqaha [d,*]

[a] Department of Information Technology, University of Gujrat, Gujrat 50700, Pakistan
[b] Department of Computer Science, University of Gujrat, Gujrat 50700, Pakistan
[c] Department of Information Technology, Division of Science & Technology, University of Education, Township Campus, Lahore 54000, Pakistan
[d] Information and Computing Technology (ICT) Division, College of Science and Engineering (CSE), Hamad Bin Khalifa University, Doha, Qatar

ARTICLE INFO

Article history:
Received 23 June 2022
Revised 29 October 2022
Accepted 5 December 2022
Available online 22 December 2022

Keywords:
Shahmukhi
Punjabi
POS tagging
Corpus annotation
Deep neural networks
ELMo
Transfer learning

ABSTRACT

Part of Speech (POS) tagging has a preliminary role in building natural language processing applications. This paper presents the development and evaluation of the first POS tagged corpus along with a Bi-directional long-short memory (BiLSTM) network based POS tagger for Shahmukhi (Western Punjabi) at this scale. A balanced corpus of 0.13 million words has been annotated which contains text from 14 different text domains. A Shahmukhi POS tagset has been devised by studying the applicability of the CLE Urdu POS tagset and tagging guidelines have also been designed for annotation. A multi-step corpus evaluation process has been employed for tagged corpus including grammar-based and n-gram based consistency evaluations. The average inter-annotator agreement for all domains is 95.35% along with an average Kappa coefficient of 0.94. The performance of the BiLSTM POS tagger has been compared with the well-known language independent TreeTagger and the Stanford POS tagger. The accuracy of the tagger has been further improved by employing transfer learning by training context-free (Word2Vec) and contextualized (ELMo) word representations on a corpus of 14.9 Shahmukhi words which has been collected from World Wide Web. The tagger performed with an f-score of 96.11 and the accuracy of 96.12%. For a morphologically-rich and low-resourced language, these POS tagging results are quite promising.

1. Introduction

Punjabi is an Indo-Aryan language and the 10th most spoken language of the world with 150 million Punjabi speakers worldwide (Ahmad et al., 2020; Simons and Fennig, 2017). It is widely spoken in Pakistan, India, Canada, USA, England and other countries with Punjabi immigrants. Being a major language in Pakistan, it is spoken by 44.15% population (PBS, 2017) of the country. It is a morphologically rich language and has been influenced by other languages such as Arabic, Urdu, Hindi, Persian, English and Sanskrit (Humayoun and Ranta, 2010; Sharma, 2016). There are various dialects of Punjabi. Majhi, Doabi, Powadhi and Malwai dialects are mainly used in Eastern Punjab whereas Pothohari, Lahandi and Multani are used in Western Punjab. However, the Majhi dialect is spoken in both Eastern and Western Punjab and is considered as the standard dialect of Punjabi (Kaur et al., 2017). The inimitable peculiarity of Punjabi is that it is written in two mutually incoherent scripts; Shahmukhi and Gurmukhi. Shahmukhi is mainly used in Western Punjab of Pakistan. It is written in Perso-Arabic script; which is right-to-left, cursive, context sensitive writing system (Malik, 2006). On the other hand, Gurmukhi is written in left-to-right syllabic script and is used in Eastern Punjab of India (Dua et al., 2012; Saini and Lehal, 2011). The two incomprehensible scripts create a wedge as most people from Western Punjab can't peruse Gurmukhi script. Similarly, a majority of Gurmukhi readers from Eastern Punjab can't understand Shahmukhi script. Interestingly, they don't have any issue to comprehend the verbal articulation of the language (Lehal, 2009).

Despite a large number of speakers, in terms of computation, Shahmukhi remains a low resourced language. However, a few works have been done for Shahmukhi in recent years which include; named entity recognition (Ahmad et al., 2020), analysis of lexico semantic relationships of nouns (Akhter et al., 2019;

* Corresponding author.
E-mail address: aalfuqaha@hbku.edu.qa (A. Al-Fuqaha).
Peer review under responsibility of King Saud University.

Production and hosting by Elsevier

https://doi.org/10.1016/j.jksuci.2022.12.004
1319-1578/

Hashmi et al., 2019a) and verbs (Hashmi et al., 2019b), vocabulary profile (Arslan et al., 2019) and stemming (Mateen et al., 2017). Although, remarkable work has been done for Gurmukhi script including; Punjabi machine translation (Malik, 2006), text summarization (Gupta and Kaur, 2016), stemming (Gupta and Lehal, 2011), text to speech (Singh and Lehal, 2006), Part of Speech (POS) tagging (Sharma and Lehal, 2011) and word sense disambiguation (Walia et al., 2019). Shahmukhi resources and techniques are quite scant as compared to Gurmukhi. Most researchers use transliteration between Shahmukhi and Gurmukhi for various tasks such as to achieve POS tags for their Shahmukhi text (Akhter et al., 2019; Hashmi et al., 2019a; Hashmi et al., 2019b; Arslan et al., 2019). The transliteration encounters various issues. Detailed analysis of the transliteration issues has been presented in Section 2.1. For low resourced Shahmukhi, POS tagging is not explored as POS tagset, POS tagged corpora and POS taggers are not available (Ahmad et al., 2020). POS tagging is an indispensable preprocessing operation for any language to address many Natural Language Processing (NLP) tasks such as speech synthesis, information extraction, machine translation, syntactic parsing and sentimental analysis etc. Therefore, there is a need to develop independent and balanced resources for Shahmukhi.

In this research, a Shahmukhi POS tagged corpus and a Bi-directional Long-Short Term Memory (BiLSTM) network based POS tagger is developed. In order to build Shahmukhi corpus, online sources are used for corpus collection which are mentioned in Table 3. Several preprocessing tasks have been performed for corpus cleaning and preparation. A corpus of 14.9 million words was collected and has been divided into 14 text domains that are shown in Table 4. For devising a POS tagset for Shahmukhi, the applicability of the CLE Urdu POS tagset (Ahmed et al., 2015) has been studied and a suitable Shahmukhi POS tagset of 36 POS classes has been devised. From 14 text domains, a balanced corpus of 0.13 million words has been extracted for annotation. Multi-tier corpus evaluation has been performed to develop a valid and accurate POS tagged corpus. The corpus evaluation process starts with completeness and correctness and followed by manual review of the tagged corpus. Further to ensure consistency, grammar-based and n-gram based consistency evaluations have been performed. Inter-annotator agreement has been computed for all domains and average agreement is 95.35% with 0.94 average Kappa value. The final version of the annotated corpus has been used for taggers training and testing. A BiLSTM neural network based POS tagger is developed and has been evaluated on the POS tagged corpus. In order to develop BiLSTM POS tagger, we performed a few experiments with various features that are discussed in Section 6.2. Transfer learning has been incorporated in the tagger training.

Two language-independent POS taggers; TreeTagger and Stanford tagger have been trained. The TreeTagger performed with an f-score of 85.11 and 85.77% accuracy whereas the Stanford tagger performed with 94.36 f-score and 94.43% accuracy. Our BiLSTM POS tagger outperformed both taggers with an f-score of 96.11 and an accuracy of 96.12%. For transliterating Shahmukhi text into Latin characters, a transliteration scheme proposed by Kamran Malik et al. (2010) has been used in all examples. The POS tagged corpus is freely available online[1]. The division of the paper is as follows. Section 2 presents a comprehensive review of literature in the field. Section 3 discusses corpus collection and preparation. In Section 4, devised POS tagset for Shahmukhi is presented and tagging guidelines are discussed. Section 5 comprises the annotation process and discusses evaluation of the corpus. Section 6 presents our BiLSTM POS tagger development and its training, other statistical POS taggers training and interpretation of taggers results. Section 7 concludes the paper by discussing findings.

2. Related work

Related work is presented in three sections; Section 2.1 presents work that has been done for Shahmukhi POS tagging, Section 2.2 presents analysis of Gurmukhi and Urdu POS tagsets and Section 2.3 discusses several POS tagging models which have been trained on Gurmukhi and Urdu. We have studied Urdu POS tagging because Shahmukhi and Urdu use identical script and grammatical similarities also exist.

2.1. Shahmukhi POS tagging

In this section, we present the literature which highlights the unavailability of Shahmukhi POS tagger. To achieve Shahmukhi POS tagged text, researchers performed POS tagging by using Gurmukhi tagger. The tagging has been done by transliterating the Shahmukhi text to Gurmukhi. Transliteration schemes for Shahmukhi and Gurmukhi have been proposed. Malik (2006) presented Shahmukhi to Gurmukhi transliteration using character mappings and dependency rules. Saini et al. (Saini et al., 2008) proposed Shahmukhi to Gurmukhi transliteration scheme and achieved accuracy of 91.37%. A Gurmukhi to Shahmukhi transliteration system was proposed by Lehal (Lehal, 2009) with 98.6% accuracy at word level. Despite of these transliteration schemes, it has been observed that these aren't used by researchers instead a software named Akhar[2] an Indic word processor has been used for Shahmukhi-Gurmukhi transliteration.

Transliteration from Shahmukhi to Gurmukhi and then back to Shahmukhi itself is a challenging task. For instance, transliterating from Shahmukhi to Gurmukhi face the issue of recognition of Shahmukhi text without diacritical marks. Shahmukhi text is normally written without short vowels which leads to potential ambiguity. For example, word رکھ *rkH* when written with short vowel zabar it becomes verb رَکھ *rakH* 'keep' but when short vowel paish is used it acts as noun رُکھ *rukH* 'tree'. This pose a great challenge to identify the right text. Multiple mappings of Shahmukhi characters in Gurmukhi exist, which increase transliteration errors. For Instance, Shahmukhi characters 'و' and 'ی' have multiple mappings in Gurmukhi. Shahmukhi character 'و' could be mapped to Gurmukhi characters '*v*' and '*o*' resulting in different Gurmukhi words (Saini and Lehal, 2011). For Gurmukhi character's Roman transliteration, we have used Gurmukhi to Roman transliteration system developed by Singh et al. (Singh and Sachan, 2019).

Similarly, when transliterating from Gurmukhi to Shahmukhi, it faces issues of difference between orthography and pronunciation, transliteration of proper noun, missing nukta (dot) symbol in Gurmukhi text and multiple character mappings (Lehal, 2009). Orthography and pronunciation makes transliteration challenging. For example, Shahmukhi character 'خ' is transformed to 'کھ' such as word خیال *xyAl* 'idea' result in word کھیال *kHyAl*. Moreover, multiple mappings create ambiguity as there are certain Gurmukhi characters which are mapped to multiple Shahmukhi characters. Gurmukhi character '*z*' could be mapped to 'ذ', 'ز', 'ظ' and 'ض'. Likewise, Gurmukhi character '*t*' has two mappings in Shahmukhi 'ط' and 'ت'. For instance, for Gurmukhi word *taakt* correct Shahmukhi word is طاقت *tAqt* 'strength' but due to multiple mappings, transliteration could result in wrong text such as تاکت, تاکط, تاقط, طاکت, طاقط, طاکط and تاقت. In Shahmukhi, some proper nouns are written using typical spellings for which general rule based mapping yields

[1] https://github.com/toqeerehsan/Shahmukhi-POS-Tagging.

[2] http://www.akhariwp.com/.

wrong text. Formation of transliteration rules for such words is challenging as general rule based mapping does not work. Further for nukta symbol, though five consonants *'sh', 'khh', 'ghh', 'z' and 'f'* have been added to basic 35 Gurmukhi characters but normally Gurmukhi users don't make distinction between nukta and without nukta character while writing such as mostly Gurmukhi character '*khh*' with nukta is replaced with character '*kh*' without nukta resulting in wrong transliterated Shahmukhi text. Moreover, there are no exact equivalent mappings for certain Shahmukhi characters in Gurmukhi script such as for 'ع' and 'ہ'. Manual cleaning of the corpus is required to address highlighted transliteration issues. A corpus based study on Lexico Semantic Relations (LSR) of Shahmukhi nouns was performed by Hashmi et al. (2019a). Shahmukhi was transliterated to Gurmukhi for POS tagging. LSR of Shahmukhi verbs has been studied by Hashmi et al. (2019b). In this work, Shahmukhi was transliterated to Gurmukhi for POS tagging. After annotation, transliteration was performed from Gurmukhi to Shahmukhi, and the corpus was manually reviewed to resolve transliteration issues.

Arslan et al. (2019) worked on the Vocabulary Profile (VP) of Shahmukhi. This was a corpus based study in which a corpus of two million was developed for selecting nouns in order to develop synsets. With respect to semantic categories the list of 1,000 nouns was developed in English. The retrieved list of nouns was translated from English to Gurmukhi and then Gurmukhi to Shahmukhi. Transliteration was used for POS tagging as POS tagger for Shahmukhi wasn't available. Transliteration issues were encountered between Gurmukhi and Shahmukhi and corpus was cleaned manually. Shahmukhi's synsets of nouns were studied in Akhter et al. (2019). Transliteration was used for POS tagging.

Named Entity Recognition (NER) and classification for Shahmukhi is studied by Ahmad et al. (Ahmad et al., 2020). They developed a corpus of 318,275 tokens. They manually tagged named entities and released the first NER corpus of Shahmukhi. They highlighted the unavailability of the Shahmukhi POS tagger.

The studies in which Shahmukhi-Gurmukhi transliteration has been performed are mentioned in Table 1. They worked on verbs, nouns and vocabulary profile of Shahmukhi and for these tasks POS tagged corpus is required in order to identify the parts of speech. These studies have used transliteration to achieve POS tagged corpus due to unavailability of Shahmukhi POS tagger. During Shahmukhi-Gurmukhi bidirectional transliteration, aforementioned transliteration issues were encountered by researchers. Manual cleaning of the corpus has been performed to develop a valid corpus which makes transliteration complex and time consuming as well. Therefore, there is need to develop independent resources for Shahmukhi. POS tagging is not only required for tasks mentioned in Table 1 and NER but it is also a pivotal task for other natural language processing applications.

2.2. Gurmukhi and Urdu POS tagsets

A few POS tagsets have been devised for Urdu (Hardie, 2003; Sajjad and Schmid, 2009; Ahmed et al., 2015) and Gurmukhi (Singh, 2008; Kumar and Josan, 2012; TDIL, 2014). NLP tasks that practice Hardie (Hardie, 2003) Urdu POS tagset and G. Singh (Singh, 2008) Gurmukhi POS tagset encounter some challenges like morpho-syntactic ambiguity and data sparseness due to a large number of tags in them. Sajjad and Schmid's (Sajjad and Schmid, 2009) Urdu POS tagset comprise inadequate analysis of verbs as there was only one tag for verb (Mukund et al., 2010). It is deliberated as purely syntactic which has paucities in morphological and functional features. These issues limits its usage to only POS task and considered unsuitable for other NLP tasks such as parsing (Abbas, 2014). Whereas, the CLE Urdu POS tagset (Ahmed et al., 2015) accuracy is considered quite sufficient, linguistically sound and covers all linguistic aspects. It also covers challenges that were created by previous tagsets during computational processing of the language. In the CLE Urdu POS tagset, special contemplation has been provided to morpho-syntactic categories (Khan et al., 2019). For instance, verbs are adequately classified into morphological POS classes; infinitive and finite verbs followed by syntactic classes of verb auxiliaries; modal, tense, aspectual and progressive.

Gurmukhi POS tagsets include reduced number of Gurmukhi POS tagset developed by TDIL[3] which was used by Kaur et al. (2014) to elevate the Gurmukhi POS tagging accuracy. Another Gurmukhi tagset was devised by Kumar and Josan (2012). In TDIL tagset, pronouns were comparatively well categorized whereas in Kumar and Josan (2012)'s tagset there was a single tag for all types of pronouns. The TDIL tagset also has separate tag for Reciprocal Pronoun (PR_RCP) for which example *apasa* 'our own' is provided in tagset documentation[4]. Shahmukhi script exhibits it similar to the CLE Urdu POS tagset and Reflexive Apna (APNA) tag is applicable. Furthermore, in the TDIL tagset separate POS class for Indefinite Pronouns (PR_PRI) has been defined for word such as kisE 'to whom'. For these words, the CLE Urdu POS tagset tags Personal Pronoun (PRP) and personal Demonstrative Pronoun (PDM) are applicable instead of PR_PRI. In TDIL tagset there were two tags for wh-words; Wh-words Pronoun (PR_PRQ) and Wh-words Demonstrative (DM_DMQ). In Kumar and Josan (2012)'s tagset, they also defined a separate tag for question words. The CLE Urdu POS tagset has well categorized pronouns classes which are acutely applicable to Shahmukhi script as wh-words or question words tags from both tagsets also act pronouns like Urdu. Whereas, subordinate conjunction, coordinate conjunctions, adverb and adjective tags in the Gurmukhi tagsets somehow could be mapped to Shahmukhi script and to some extent are identical to the CLE Urdu POS tagset as well.

Kumar and Josan's (Kumar and Josan, 2012) tagset, includes four POS classes for nouns; Common Noun (NN), Proper Noun (NNP), Compound (NNC) and Compound Proper Noun (NNPC). Whereas, the TDIL tagset categorises noun into Common Noun (N_NN), Proper Noun (N_NNP) and Location Noun (N_NST). From these both tagsets, common and proper noun tags are applicable to Shahmukhi. Although, the N_NST tag in TDIL syntactically behaves like common noun in Shahmukhi similar to CLE Urdu POS tagset. In Gurmukhi script some lexical items are joined and behave as single lexical item whereas in Urdu and Shahmukhi they are considered as separate lexical items. For instance, the word *mUm batI* 'candle' is written joined in Gurmukhi and for compound nouns like these (Kumar and Josan, 2012)'s Gurmukhi POS tagset assigns NNC tag. In Shahmukhi and Urdu, we can't join these words and common noun tag is more applicable to them as word segmentation is identical among Urdu and Shahmukhi. Correspondingly for verbs and auxiliaries, similar challenges are opposed. As *hUyE gA* 'will be' is written as separate lexical items in Shahmukhi and Urdu and verb and auxiliary tags are assigned to them respectively. In Gurmukhi, they are written as single lexical item and a single verb tag is assigned. To analyze examples provided in Gurmukhi tagsets guidelines, we used Google translator for translation.

For symbols, Kumar and Josan's (Kumar and Josan, 2012) tagset comprises too many tags. The Main Verb (VBM) category also has been subcategorized into tense and person POS. For tagging, tense and person POS tags are used in conjunction with VBM tag. Consequently, more than two tags are used in conjunction which increase the size of the tagset. Both tagsets also have defined separate tags for foreign fragments residual and unknown words. The CLE Urdu POS tagset also provides POS class for Foreign Fragment

[3] http://punjabipos.learnpunjabi.org/TagSet.aspx.

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

Table 1
Overview of Shahmukhi POS tagging by using Gurmukhi POS tagger after transliterating Shahmukhi to Gurmukhi.

References	Transliteration	VP	Verb LSR	Noun LSR	Noun synsets	Gurmukhi POS Tagger	Shahmukhi POS Tagger
Hashmi et al. (2019a)	✓	X	X	✓	X	✓	X
Akhter et al. (2019)	✓	X	X	X	✓	✓	X
Hashmi et al. (2019b)	✓	X	✓	X	X	✓	X
Arslan et al. (2019)	✓	✓	X	X	X	✓	X

(FF) but instead of unknown words tag, it provides obvious POS classes. Conclusively, the script barrier and lack of documented work, it's hard to study the applicability of Gurmukhi POS tagsets for Shahmukhi in more details. Whereas, the script of Urdu and Shahmukhi is same and both share a large vocabulary (Bhurgari, 2007; Hasan et al., 2015). Grammatical similarities also exist including morphological richness, case markers, complex predicate structures and copula constructions. In future, Urdu grammar could also help in syntactic analysis of Shahmukhi due to syntactic structure similarities and applicability. By considering all discussed parameters, the CLE Urdu POS tagset has been considered as benchmark for devising a Shahmukhi POS tagset.

2.3. Existing Gurmukhi and Urdu POS taggers

POS taggers have been developed by using rule-base, probabilistic, hybrid, machine and deep learning approaches. Stochastic taggers are based on techniques like Conditional Random Field (CRF) (Lafferty et al., 2001), Hidden Markov Model (HMM) (Charniak et al., 1993), Support Vector Machines (SVM) (Giménez and Marquez, 2004), Maximum Entropy (ME) (Ratnaparkhi, 1996), and Decision Tree (DT) (Schmid, 2013).

M. Kaur et al. (Kaur et al., 2014) improved Gurmukhi POS tagger performance by using a reduced tagset and an HMM based model. A corpus was collected from cyberspace which contained 42,000 words. They used a tagset which was proposed by Technical Development of Indian Languages (TDIL)[5] consisting 36 tags. As large tagset used by Sharma and Lehal (2011) resulted in data sparseness problem. The proposed model reported 92–95% accuracy.

A statistical bi-gram model for Gurmukhi POS tagging was developed by Mittal et al. (2014). They used a tagset proposed by TDIL. Online resources were used to collect the corpus of 2,400 sentences consisting of 10,000 words. They achieved 92.16% POS accuracy without unknown words. A hybrid rule-based and CRF based approach was proposed by Sharma (2016) for Gurmukhi POS tagging. A tagset of 36 classes was used as proposed by TDIL. A corpus of 28,000 to 42,000 words was used. The tagger performed with 92% accuracy. An SVM based tagger for Gurmukhi was developed by Kumar and Josan (2016). A Punjabi POS tagset containing 38 classes was proposed by Kumar and Josan (2012). A corpus of 54,000 words was used for training and testing the accuracy of the tagger. An average accuracy of 89.90% was reported. A statistical POS tagger for Urdu was proposed by Anwar et al. (Anwar et al., 2007). The tagger was based on n-gram Markov and subsequently backoff model. EMILLE corpus was used for training and testing. Experiments were performed using two tagsets, the first tagset contained 250 tags and the second one had 90 tags. Backoff model with small tagset gave a best accuracy of 95%.

Sajjad and Schmid (Sajjad and Schmid, 2009) experimented with four state of the art probabilistic POS taggers for Urdu: TnT tagger, TreeTagger, SVM tagger and Random Forest (RF) tagger and compared their performances. A corpus of 110,000 words was collected from cyberspace. A tagset with 42 tags was used for annotation. It was concluded that SVM tagger performed better which yielded an accuracy of 95.66%.

For Urdu, Ahmed et al. (Ahmed et al., 2015) proposed a POS tagset. It has 12 major categories with subdivisions ensuing in 35 syntactic classes. The tagset was used to annotate 100 thousand words from the CLE Urdu Digest Corpus (Urooj et al., 2014) and 96.8% accuracy was reported which is quiet promising for a morphologically rich language.

A CRF based POS tagger for Urdu was proposed by Khan et al. (2019) and results were evaluated against SVM. Two benchmark datasets were used; CLE dataset (Urooj et al., 2014) and BJ dataset (Jawaid et al., 2014). Two tagsets were used one of 35 syntactic classes and other one of 37 syntactic classes. The reported accuracy for CLE and BJ datasets were 86.95% and 93.56% respectively. The work presented in Khan et al. (2019) has been further extended by Khan et al. (Khan et al., 2019). In this research, the effectiveness of machine and deep learning approaches was studied. The models were evaluated on two benchmark datasets: CLE dataset and BJ dataset. The models which were evaluated include; CRF, SVM, HMM, n-gram Markov model, and two variants of Recurrent Neural Network (RNN). These two variants of RNN comprise of simple LSTM-RNN and LSTM-RNN with CRF. It was yielded by experiments that on CLE dataset CRF based model performed better with 83.52% accuracy. In contrary, on BJ dataset LSTM-RNN model performed better by reporting 88.7% accuracy. Nasim et al. (Nasim et al., 2020) used two state-of-the-art taggers: CRF and BiLSTM with CRF. These two models mainly were used for sequential tagging. Both models attained an accuracy of 96%.

Ehsan and Butt (Ehsan and Butt, 2020) worked on dependency parsing for Urdu. They presented a comparative study of training two dependency parsers; BiLSTM parser (Kiperwasser and Goldberg, 2016) and MaltParser (Nivre et al., 2007). Both parsers were trained on two dependency treebanks of Urdu; CLE-UTB (Ehsan and Hussain, 2020; Ehsan and Hussain, 2019; Ehsan, 2022) and HUTB-UTB[6]. The CLE-UTB used a tagset of 40 tags and a corpus of 148,000 tokens having 7,854 sentences. They further developed a POS tagger which was based on BiLSTM networks. The tagger performed with a best accuracy of 96.3% on CLE-UTB. The POS tagging is also helpful for training shallow parsers (Ehsan et al., 2022). Table 2 summarizes the POS tagging approaches that are presented for Gurmukhi and Urdu.

3. Shahmukhi Corpus

Fig. 1 shows the methodology of the development of the Shahmukhi POS tagged corpus and Neural POS tagger. At first step, a corpus containing 14.9 million Shahmukhi words has been collected from Word Wide Web. Then corpus has been normalized by performing Unicode normalization and cleaning. A smaller version of the corpus has been selected for annotation with respect to 14 text domains. The applicability of an Urdu POS tagset has been studied and a tagset has been proposed for Shahmukhi. Annotation

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

[6] https://github.com/UniversalDependencies/UDUrdu-UDTB.

Table 2
Summary of POS tagging models, corpora and results for Gurmukhi and Urdu.

References	Approach/ Model	Dataset	Tagset	Language	Results/Accuracy
(Kaur et al., 2014)	Statistical:HMM	42,000 words	TDIL tagset of 36 tags	Gurmukhi	92–95% accuracy
(Mittal et al., 2014)	Statistical:Bi-gram model	2400 sentences having 10,000 tokens	36 tags proposed by TDIL tagset	Gurmukhi	92.16% accuracy
(Sharma, 2016)	Hybrid:Rule-based and statistical:CRF	Corpus of 28,000 to 42,000 words	36 classes	Gurmukhi	92% accuracy
(Kumar and Josan, 2016)	Statistical: SVM	Corpus of 54,000 words	38 tags	Gurmukhi	89.90% accuracy
(Anwar et al., 2007)	Statistical:n-gram Markov model	EMILLE Corpus: Training corpus having 1000 words	Two tagset one with 250 Tags and other of 90 tags	Urdu	95% accuracy
(Sajjad and Schmid, 2009)	Statistical:RF Tagger, TreeTagger, SVM and TnT tagger	Corpus of 1,10,000 collected from cyberspace	42 tags	Urdu	SVM tagger performed best with 95.66% accuracy
(Ahmed et al., 2015)	Statistical	CLE Urdu Digest corpus of 100 K words	35 tags	Urdu	96.8% accuracy
(Khan et al., 2019)	Statistical: CRF evaluated	CLE dataset and BJ dataset	Two tagsets one with 35 tags and one with 37 tags	Urdu	86.95% accuracy on CLE dataset and 93.56% accuracy on BJ dataset
(Khan et al., 2019).	Statistical: CRF, HMM, SVM RNN: LSTM and LSTM with CRF	CLE dataset and BJ dataset	Two tagsets: CLE tagset and Sajjad tagset	Urdu	For CLE dataset CRF model performed better with 83.52% accuracy and on BJ dataset LSTM-RNN model performed better with 88.7% accuracy
(Nasim et al., 2020)	Statistical: CRF RNN: BiLSTM CRF	Dataset by(Jawaid et al., 2014)	Sajjad tagset 42 tags	Urdu	96% accuracy
(Ehsan and Butt, 2020)	RNN: BiLSTM tagger	Corpus of 1,48,000 tokens having 7,854 sentences	40 tags	Urdu	96.3% accuracy

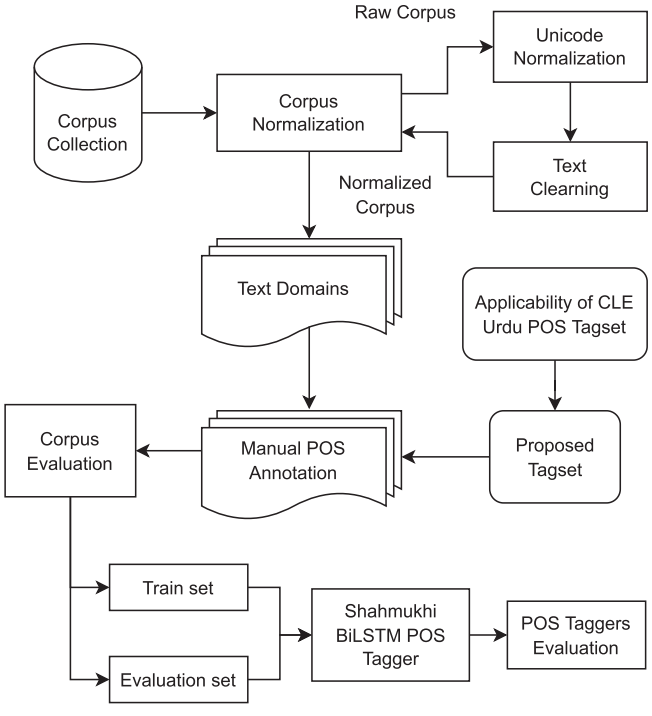

Fig. 1. Methodology of the development of the Shahmukhi POS tagged corpus and Neural POS tagger.

Guidelines were prepared for manual annotation. Several evaluation techniques have been employed to ensure a valid annotated corpus. Finally, the annotated corpus has been divided into train and evaluation sets to train neural and other statistical POS taggers. In this section, Section 3.1 discusses corpus collection whereas in Section 3.2 corpus preparation process is presented.

3.1. Corpus collection

Corpora are essential resources for NLP tasks. To build benchmark corpora for Western and Asian languages, the online sources including; News, Wikipedia, Emille dataset, novels, short stories and articles have been widely acknowledged as primary sources (Anwar et al., 2007; Hashmi et al., 2019a). In this work, a Shahmukhi corpus of significant size has been collected. There are very few Shahmukhi textual sources on the Internet, as mostly online Shahmukhi source have data in graphic format which couldn't be utilized for language processing. The Shahmukhi text corpus has been collected from several online available sources which are shown in Table 3.

Among these online sources *Wichaar* and *Bhulekha* are news websites. *Punjabi Stories* comprises Shahmukhi short stories written by a number of writers. Well-known sources *Wikipedia* and *EMILLE* significantly contribute to corpus size. The text crawling has been performed using Cyotek webcopy[7].

3.2. Corpus preparation

For corpus preparation, first step was to parse the crawled data and acquire substantial amount of raw Shahmukhi text. As the data crawled from the websites was in the html format. A corpus of almost 14.9 million tokens and 797 thousand sentences was obtained from sources in UTF-8 format. Table 3 shows detail statistics of the collected corpora with respect to sources. Further, following tasks have been performed for corpus preparation:

[7] https://www.cyotek.com/cyotek-webcopy.

Table 3
Online sources used for corpus collection and number of tokens and sentences each source contribute to the corpus.

Sr.#	Sources	No. of Tokens	No. of Sentences
1	Wichaar[a]	2,239,974	126,069
2	Bhulekha[b]	937,619	28,051
3	Punjabi Stories[c]	1,718,177	105,031
4	EMILLE[d]	5,218,157	315,021
5	Punjabi Wikipedia[e]	4,883,034	223,762
	Total	14,996,961	797,934

[a] http://www.wichaar.com/.
[b] https://bhulekhatv.com/
[c] http://www.punjabikahani.punjabi-kavita.com/Punjabi-StoriesShahmukhi.php.
[d] https://cqpweb.lancs.ac.uk/emillewpun/.
[e] https://pnb.wikipedia.org/wiki/.

3.2.1. Unicode normalization

The Shahmukhi alphabets are not standardized yet. Consequently, the collected corpus from online sources comprise of Unicodes from various languages. Accordingly, it was necessary to map the corpus to unified Unicodes for consistency throughout the corpus. Shahmukhi is generally written by using Urdu alphabets (Bhurgari, 2007; Hasan et al., 2015). Therefore, the corpus has been charted to Unicodes which exist in an Urdu keyboard released by CLE[8]. A utility was built to generate report of Unicodes that don't exist in the CLE keyboard and necessary mappings were made. For example, 'آ' (0627 + 0653) and 'ك' (0643) has been mapped to 'آ' (0622) and 'ک' (06A0) respectively. Though some characters which don't had particular mappings were removed such as ̂, Î, ‘and Ý etc.

3.2.2. Cleaning

For corpus cleaning, word tokenization has been overlooked and sentence segmentation has been performed. Symbols like ٪, ۔,], x, ؟ and + etc. have been handled as separate tokens. The word tokenization was performed manually during the POS tagging. Non-printable characters were removed and all duplicated files were discarded.

3.2.3. Text domains/genres

Based on the contents, the corpus of 14.9 million words has been divided into 14 text domains. Text domains with detailed statistics are shown in Table 4. Text data available in *EMILLE* and *Punjabi Wikipedia* had plain text so both are handled as separate text domains and labeled as *Emille Dataset* and *Pnb Wikipedia* respectively.

4. POS tagging

POS tagging is complementary element for any language. For formulating Shahmukhi POS tagset and tagging guidelines, the CLE Urdu POS tagset has been considered as benchmark and its applicability has been studied. The CLE Urdu POS tagset is considered linguistically sound and covers all linguistic aspects as discussed in **Section 2.2**. The following sections present devised Shahmukhi POS tagset, tagging guidelines and annotation challenges that are faced during annotation of Shahmukhi.

4.1. POS tagset

The devised Shahmukhi POS tagset entails 12 major POS categories with 36 subcategories. The major categories remain same as of the CLE Urdu POS tagset. However, one additional tag Pronominal Suffixes (PRX) in the pronoun class has been added. Table 5 shows the devised POS tagset with respect to categories. Noun category has been divided into two subcategories common and proper noun whereas noun modifiers incorporate six subcategories; adjectives, quantifier, cardinal, ordinal, fraction and multiplicative. Verb and auxiliary verbs evince two and four subcategories respectively. Pronouns category encompasses maximum subcategories than other POS categories as there are eight subcategories under pronoun. Tagset comprises a single tag for each category of residual and interjection. Adverb, symbol, particle and adposition each encompass two categories. Conjunction have four subcategories; coordinate, subordinate, pre-sentential and SCKar. Conclusively, tagset covers all morpho-syntactic categories of Shahmukhi.

4.2. Tagging guidelines

The following sections give a comprehensive analysis of POS tagset classes with the help of Shahmukhi examples.

4.2.1. Verbs

Verb is subcategorized into Main Verb Infinitive (VBI) and Main Verb Finite (VBF). VBI are verbs with infinitive forms such as *khAnA*

(1)

جدوں	بچہ	پنج	سال	دا	ہوندا	اے	تے	اوھدے	ذھن	وِچ
jidUN	baca	panj	sAl	dA	hOndA	aE	tE	oUhdE	zihn	vic
when	child.M.Sg	five	year	case.Gen.M.Sg	be.Pres.3.M.Sg	be.Pres.3.Sg	then	pron.3.Sg	mind	case.Loc
NN	NN	CD	NN	PSP	VBF	AUXT	SC	PRS	NN	PSP

ڈھائی	ہزار	الفاظ	ذخیرہ	کرن	دی	گُنجائش	ہوندی	اے
DHAI	hzAr	alfAz	z2xIrh	kran	dI	gunjAS	hOndI	aE
2.5	thousand	words	save	do	case.Gen.F.Sg	capacity	be.Pres.3.F.Sg	be.Pres.3.Sg
FR	CD	NN	NN	VBI	PSP	NN	VBF	AUXT

'When a child reaches the age of five then he has the capacity to memorize two and a half thousand words in his mind.'

'to eat' and *calnA* 'to walk'. Verb forms that show tense are tagged as VBF. In (1) *hOndA* 'happen' is VBF and *kran* 'to do' is VBI.

[8] http://www.cle.org.pk/software/localization/keyboards/CRULPphonetickb2Lv1.0.htmls.

4.2.2. Auxiliary verbs

Auxiliaries are subcategorized into Aspectual (AUXA), Progressive (AUXP), Modal (AUXM) and Tense (AUXT). AUXA refers to the verb that appears after the main verb like *jAndI* 'go' in (6) and *cukE* 'done' in (2). AUXP marks something which is in progress like *rhI* 'becoming' in (3) and AUXM indicates capability, willingness or possibility such as *sakdE* 'can' in (4). AUXT shows the tense of the action as *aE* 'is' in (3).

4.2.3. Nouns

Noun class has been divided into two subcategories: Common Noun (NN) and Proper Noun (NNP). Examples of common nouns are *kuRyAN* 'girls', *bOlI* 'language' and *dunyA* 'world'. Nouns that show adverbial nature are called spacio-temporal nouns which refers to space and time such as *aIthE* 'here', *hun* 'now', *jidUN* 'when' and *bAhr* 'outside' are also tagged as NN such as in (5) *aIthE* is tagged as NN. Named entities and abbreviations are assigned tag NNP. In (6) *panjAbI* 'Punjabi' and *panjAb* 'Punjab' are NNP.

(2)

اسیں	اے	کتاب	پڑھ	چکے	آں
asIN	aE	kitAb	paRH	cukE	aN
pron.1.Pl	pron.2.Sg	book.F.Sg	read	do.Perf.F.Pl	be.Pres.3.Pl
PRP	PDM	NN	VBF	AUXA	AUXT

'We have read this book.'

(3)

پوری	دنیا	اک	پنڈ	بندی	جا	رہی	اے
pUrI	dunyA	ak	pinD	bandI	jA	rhI	aE
whole	world.F.Sg	one	village.M.Sg	become.Pres.3.F.Sg	go	live.F.Sg	be.Pres.3.Sg
JJ	NN	CD	NN	VBF	AUXA	AUXP	AUXT

'The whole world is becoming a village.'

(4)

تسیں	جا	سکدے	او
tusIN	jA	sakdE	aU
pron.2.Pl	go	can.Pres.M.Pl	be.Pres.3.Pl
PRP	VBF	AUXM	AUXT

'You can go.'

(5)

او	ایتھے	رہندا	اے
oU	aIthE	rehndA	aE
pron.3.Sg	here	live.Pres.3.M.Sg	be.Pres.3.Sg
PRP	NN	VBF	AUXT

'He lives here.'

(6)

پنجابی	پنجاب	وچ	بولی	جاندی	اے
panjAbI	panjAb	vic	bOlI	jAndI	aE
Punjabi.F.Sg	Punjab.M.Sg	case.Loc	speak	go.Pres.3.F.Sg	be.Pres.3.Sg
NNP	NNP	PSP	VBF	AUXA	AUXT

'Punjabi is spoken in Punjab.'

4.2.4. Pronouns

In the CLE Urdu POS tagset, pronouns are divided into seven categories based on their syntactic nature; Personal Pronoun (PRP), Demonstrative Pronoun (PDM), Relative Demonstrative Pronoun (PRD), Relative Personal Pronoun (PRR), Reflexive Pronoun (PRF) and Reflexive Apna Pronoun (APNA). However, we have added additional subcategory in pronoun POS classes; Pronominal Suffixes (PRX). PRX are objective and possessive pronouns to nouns. In Shahmukhi PRX are 'jE', 'nI', 'nE', 'sU', 'vE' and 'aE'. In (10) 'jE' is PRX. PRX words sometimes act as auxiliary and take AUXT tag. (Butt, 2007) have discussed in detail about the role of pronominal suffixes in Punjabi.

PRP appears as the head of a noun phrase and replaces the noun such as *asIN* 'we' and *tusIN* 'you' are tagged as PRP in (2) and (4) respectively. PDM appears before a noun as its specifier such as *ah* 'this' in (7). PRS spectacles the possession such as *sAdA* 'our' and *aIsdA* 'his'. The word *tuhADA* 'yours' is PRS in (9). PRR and PRD refer to the relative pronouns. In (9) *vargA* 'like' is PRR. PRD refers to the noun which is immediately followed by it like *jEs* 'which' in (8). APNA refers to the self relation with the noun. PRF refers to oneself. The words *apnA* 'own' in (7) and *apE* 'himself' in (11) are APNA and PRF respectively. Similar to Urdu, in Shahmukhi interrogative words also act as pronouns, quantifiers and adverbs such as *kUn* 'who', *kitnA* 'how much' and *kI* 'what' respectively.

(10)

اصل	وچ	کیہہ	چاہندا	جے
as3al	vic	kIhh	cAhndA	jE
actual	case.Loc	what	want.Pres.3.Sg	be.Pres.3.Sg
NN	PSP	RB	VBF	PRX

'What does (he) actually want?'

(11)

تساں	آپے	آکھیا	سی
tusAN	apE	akhyA	sI
pron.2.Pl	yourself	say.Perf.M.Sg	be.Perf.M.Sg
PRP	PRF	VBF	AUXT

'You said it by yourself.'

4.2.5. Nominal modifiers

Nominal modifiers give information about nouns. They comprise Adjectives (JJ) such as word *pichlE* 'previous' in (17), Quantifiers (Q): *sabh kujh* 'everything' in (12), Cardinal (CD): *panj* 'five' in (1), Ordinal (OD): *pahlI* 'first' in (14), Fraction (FR): *DHAI* in (1) and multiplicative (QM): *dUgnI* 'double' in (13).

(7)

آہ	چڑھدا	پنجاب	وی	میرا	اپنا	ہے
ah	caRHdA	panjAb	vI	mErA	apnA	hE
pron.2.Sg	rising	Punjab	also	pron.1.Sg	mine	be.Pres.3.Sg
PDM	JJ	NNP	PRT	PRS	APNA	VBF

'This rising Punjab is also my own.'

(8)

جیس	دن	اسی	لاہور	گئے	تے	وآپس	آئے
jEs	din	asI	lAhOr	gyE	tE	vaps	ayE
that	day.M.Sg	pron.1.Pl	Lahore	go.Perf.M.Pl	and	back	come.Perf.M.Pl
PRD	NN	PRP	NNP	VBF	CC	NN	VBF

'The day we went to Lahore and came back.'

(9)

سگوں	تہاڈا	انجام	وی	اوس	بندے	ورگا	ہووے	گا
sagUN	tuhADA	anjAm	vI	oUs	bandE	vargA	hOvE	gA
so	pron.2.Sg	end	also	pron.3.Sg	person.M.Sg	like	be.Pres.Sg	be.Fut.M.Sg
SCP	PRS	NN	PRT	PDM	NN	PRR	VBF	AUXT

'Though your end will also be like that person.'

(12)

فیر	اوہ	سبھ	کجھ	مٹی	دا	ڈھیر	بن	گیا
fEr	oUh	sabh	kujh	mitI	dA	dhEr	ban	gyA
then	pron.3.Sg	all	some	sand.F.Sg	case.Gen.M.Sg	heap.M.Sg	become	go.Perf.M.Sg
RB	PRP	Q	Q	NN	PSP	NN	VBF	AUXA

'Then that everything became a heap of soil.'

(13)

جدکہ	اصلی	گنتی	اس	توں	دوگنی	اے
jidukE	as3lI	gintI	as	tUN	dUgnI	aE
while	real	count.F.Sg	pron.Sg	case.Com	double.F.Sg	be.Pres.Sg
SCP	JJ	NN	PRP	PSP	QM	VBF

'While the actual count is double than that.'

(14)

اے	پہلی	وار	سی
aE	pahlI	vAr	sI
this	first	time	be.Perf.F.Sg
PRP	OD	NN	VBF

'This was first time.'

the case. In both languages, case markers normally appear after noun or pronoun and act in similar manner (Ahmed, 2007; Butt and King, 2003). (Khan, 2009) worked on case system of South Asian languages and mentioned Punjabi case markers; nominative, ergative *nE*, accusative *nUN*, dative *nUN*, instrument *nAl*, ablative *tUN*, genitive *dI* and locative *vic*. In Urdu, case markers appear as separate lexical items and the CLE Urdu POS tagset marks them as PSP. After analyzing the identical behavior of Shahmukhi case markers, same guidelines have been adopted for Shahmukhi tagging. Examples of PSP are *vic* 'in' (locative case marker) in (1)

(15)

کسے	پکھیرو	نوں	بلا	وجہ	نہیں	مارنا
kisE	pkHErU	nuN	bilA	vjah	nhEN	mArnA
pron.3.Sg	bird.M.Sg	case.Acc	without	reason.F.Sg	not	kill.Inf.M.Sg
PDM	NN	PSP	PRE	NN	NEG	VBI

'Do not kill a bird without a reason.'

4.2.6. Adverbs

Adverbs are divided into Common Adverb (RB) and Negative Adverb (NEG). RB examples are *fTAfaT* 'immediately', *sukhyAN* 'easily' and *lagpag* 'almost'. The word *fEr* 'then' is RB in (12). NEG marks negations such as *nhEN* 'not' in (15).

4.2.7. Adpositions

Adpositions have two subcategories; Preposition (PRE) and Postposition (PSP). Shahmukhi PRE examples are: *bGEr* 'without', *sivAE* 'except' and *fI* 'per'. In (15) *bilA* 'without' is PRE. Indo-Aryan languages, Urdu and Punjabi use clitics and postpositions to mark and *tUN* 'from' (ablative case marker) in (13).

4.2.8. Conjunctions

Conjunctions are divided into four categories. Subordinate Conjunction (SC) comprises word that joins the dependent clause with the independent/parent clause. Whereas, Coordinate Conjunction (CC) joins two independent clauses such as *tE* 'and/then' in (8) is CC and in (1) is SC. Pre-sentential Conjunction (SCP) mark words which appear at the start of a sentence for instance *jidukh* 'while' in 13. Subordinating Conjunction Kar (SCK) appears at the end of non-finite clause such as *kE* in (16).

(16)

آهو	!	انہے	مدد	کر	کے	مثال	قائم	کر	دتی
ahO	!	onhE	madad	kar	kE	misAl	qAEm	kar	ditI
Yes	!	pron.3.Sg	help	do	case.Gen	example.F.Sg	set	do	give.Perf.F.Sg
INJ	PU	PRP	NN	VBF	SCK	NN	JJ	VBF	AUXA

'Yes! He set an example by helping.'

4.2.9. Interjection

It is an abrupt remark that expresses an impulsive feeling or reaction. It mostly transpires at the start of the sentence such as *ahO* 'yes' in (16). Multiword words such as *mASA Allah* 'God has willed it', *subhAn Allah* 'glory be to God' and *xOS2 amdId* 'welcome' are assigned a single tag INJ.

4.2.10. Particles

Particles are subcategorized into general Particle (PRT) and a language specific particle (VALA). Both particle subclasses of the CLE Urdu POS tagset are applicable to Shahmukhi script as well. In (7) *vI* 'also' is PRT. VALA particle usually appears after noun, pronoun, adjectives and infinitive verbs. In (17) *vAlE* is VALA which appears after possessive pronoun *tuhADE* 'yours' and refers to possession. Whereas, when it is used after VBI it corresponds to an action that is about to start.

4.2.12. Residual

Residual has a single tag Foreign Fragment (FF) for all words from foreign languages such as in (18) 'donate' from English language is tagged as FF. Though in Shahmukhi corpus, words from Urdu language aren't considered as FF and are tagged with the same tagset because both languages have identical script and share lots of vocabulary. Syntactic structure of sentences is also similar. Hence, same tagset is applicable for both languages.

4.3. Annotation challenges

The following sections discuss some annotation challenges that are faced during annotation of the Shahmukhi.

4.3.1. Word segmentation

Shahmukhi script uses Perso-Arabic script which is cursive in nature. Due to this feature, word segmentation problems arise.

(17)

پچھلے	دو	کمرے	تہاڈے	والے	ہن
pichlE	dO	kamrE	tuhADE	vAlE	hen
last	two	room.M.Pl	pron.2.Pl	yours	be.Pres.M.Pl
JJ	CD	NN	PRS	VALA	VBF

'The last two rooms were yours.'

4.2.11. Symbols

Symbols have two subcategories; Punctuation (PU) and Common Symbols (SYM). In (18) parenthesis () and sentence end marker '

-

' are PU SYM comprises symbols like ٪ ، ء and x etc.

Space omission and space insertion are two major issues in Shahmukhi script. Space insertion mostly occurs when space is inserted between compound words when they are supposed to appear as single lexical item. For instance, ناول کار *nAvalkAr* 'novelist' is correct compound word. if segments aren't joined then space insertion issue prompts and consequently ناول *nAval* 'novel' and کار *kAr* 'car' give different meanings. Some Shahmukhi writers use

(18)

انھے	اپنی	ساری	زمین	دان	(	donate	)	کر	دتی	۔
unhE	apnI	sArI	zmIn	dAn	(	donate	)	kar	dItI	.
pron.3.Sg	his	all	land.F.Sg	donate	(	donate	)	do	give.Perf.F.Sg	period
PRP	APNA	JJ	NN	NN	PU	FF	PU	VBF	AUXA	PU

'He donated all of his land.'

dash between compound words to join them but its not a common practice. The tokenization has been performed manually during the POS annotation. We have used Zero Width Non-Joiner (ZWNJ) to join the word segments in annotation.

4.3.2. Complex predicates

Structure in which noun, adjective and quantifier occur in verb phrase followed by a light verb is known as complex predicate. Shahmukhi and Urdu have same structure of complex predicate as Urdu complex predicate structure also comprise noun + light verb, adjective + light verb and quantifier + light verb (Butt, 1995). In Shahmukhi and Urdu, words that are followed by a light verb to make complex predicate structure turned into verb when translated into English. For example, (19) shows noun + light verb complex predicate structure. Word *yAd* 'memorize' is although NN in Shahmukhi followed by a light verb *kar* but after translating it into English, it becomes verb 'memorized'.

(19)

بچے	نے	سبق	یاد	کر	لیا
bacE	nE	sabaq	yAd	kar	lyA
child.M.Sg	case.Erg	lessen.M.Sg	memorize	do	take.Perf.M.Sg
NN	PSP	NN	NN	VBF	AUXA

'The child memorized the lesson.'

4.3.3. Verb phrases acting as adjective

Verb phrases act as adjectives frequently as they appear in place of adjective giving additional information about noun such as in (20) *caRhdE hUyE* 'rising' is verb phrase which is acting as adjective to noun *sUrj* 'sun'. In Urdu, verb phrases also act as adjective (Sajjad, 2007). We decided to tag them as verb phrase independent of noun because it consists of multiple tokens. Therefore, in (20) verb phrase *caRhdE hUyE* 'rising' is tagged VBF and AUXA respectively instead of JJ.

(20)

چڑھدے	ہوے	سورج	نوں	ویکھیا	جا	سکدا	اے
caRhdE	hUyE	sUrj	nUN	vIkhyA	jA	sakdA	aE
rising	be.Perf.M.Pl	sun.M.Sg	case.Acc	see.Perf.M.Sg	go	can.Pres.3.M.Sg	be.Pres.Sg
VBF	AUXA	NN	PSP	VBF	AUXA	AUXM	AUXT

'The rising sun can be seen.'

4.3.4. Lexical ambiguity

For POS tagging, the contextual information of word is essential as a word may have different tags based on its context. Same tag couldn't be assigned to a word in every context, this issue is known as lexical ambiguity. For example, word اے *aE* takes tags NN, NNP, VBF, AUXA, AUXT, PRP and PDM as well. Such as in 5 its tag is AUXT, in 3 it is PDM, in 13 VBF and in 14 it is PRP. In order to address lexical ambiguity the context of the word is determined and it is analyzed what tag should it carry based on that particular context.

5. Tagging evaluation

For annotation, a balanced portion of 130,003 words was randomly extracted from a corpus of 14.9 million words. This extraction has been performed in a way that it covers data from all 14 domains. Detailed statistics of the selected tokens and sentences against each domain are shown in Table 4. The annotation of the selected corpus has been performed manually. During tagging, word segmentation issues were encountered which were manually resolved. We used ZWNJ between word segments to handle extra space problems. The corpus has been annotated according to guidelines provided in Section 4. Fig. 2 shows the process of annotation evaluation.

After manual annotation of the corpus, the next step was the evaluation of the corpus. For manually annotated corpora, corpus evaluation is always an imperative and even critical concern due to elusiveness in the manually annotated corpus. A multi-step corpus evaluation has been performed. At first, we performed completeness and correctness followed by the manual review of the whole annotated corpus. Moreover, the inter-annotator agreement has been computed against each domain. Further for consistency, grammar-based and n-gram based consistency evaluations have been performed. The following sections describes corpus evaluation processes in detail.

Table 4
Text domains with total number of tokens and sentences and correspondingly selected number of tokens and sentences extracted for tagging.

Sr.#	Domains	No. of Tokens	No. of Sentences	Selected Tokens	Selected Sentences
1	Artists	80,208	3,857	11,988	519
2	Business and Economy	58,819	2,263	8,629	310
3	Comics	45,713	2,108	5,810	307
4	News	956,403	29,357	17,297	576
5	Sports	45,080	1,627	5,966	217
6	Short Stories	2,527,736	162,214	12,894	859
7	Science and Technology	36,449	1,316	5,311	206
8	Women	47,512	2,076	6,415	293
9	Novels	96,482	7,087	6,810	402
10	Punjabi Language	112,631	4,588	17,058	695
11	Articles	550,541	27,441	11,087	532
12	Classics	338,196	15,217	12,663	518
13	Emille Dataset	5,218,157	315,021	3,734	208
14	Pnb Wikipedia	4,883,034	223,762	4,341	204
	Total	14,996,961	797,934	130,003	5,846

Table 5
Devised Shahmukhi POS tagset.

Sr.#	Main Category	Sub-categories	POS Tag
1	Noun	Common noun	NN
		Proper noun	NNP
2	Verb	Main Verb Infinitive	VBI
		Main Verb Finite	VBF
3	Auxiliary Verbs	Aspectual	AUXA
		Model	AUXM
		Progressive	AUXP
		Tense	AUXT
4	Pronoun	Personal	PRP
		Demonstrative	PDM
		Possessive	PRS
		Relative demonstartive	PRD
		Relative personal	PRR
		Reflexive	PRF
		Reflexive Apna	APNA
		Pronominal suffixes	PRX
5	Nominal Modifiers	Adjective	JJ
		Quantifier	Q
		Cardinal	CD
		Ordinal	OD
		Fraction	FR
		Multiplicative	QM
6	Adverb	Common	RB
		Negation	NEG
7	Adposition	Preposition	PRE
		Postposition	PSP
8	Conjunction	Coordinate conjunction	CC
		Subordinate conjunction	SC
		Subordinate conjunction kar	SCK
		Pre-sentential	SCP
9	Interjection	Interjection	INJ
10	Particle	Common	PRT
		Vala	VALA
11	Symbol	Common	SYM
		Punctuation	PU
12	Residual	Foreign Fragment	FF

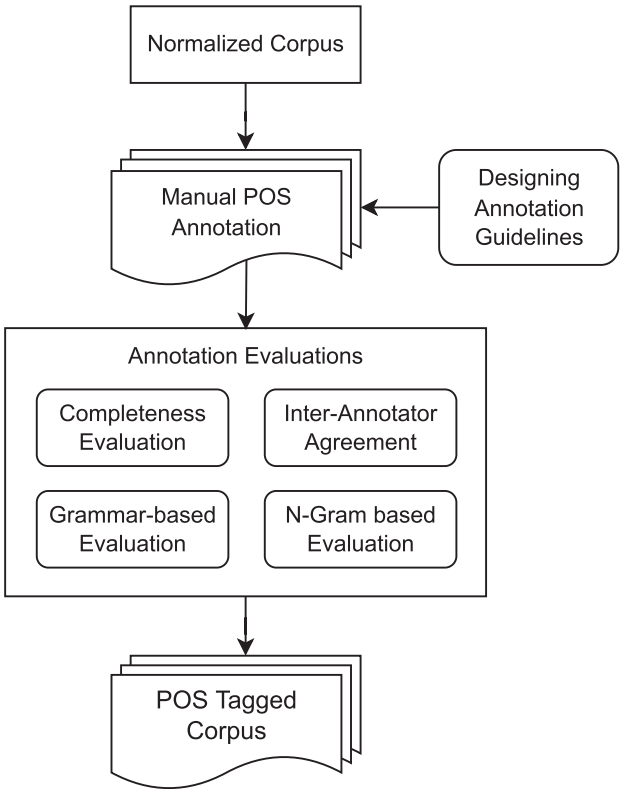

Fig. 2. Annotation evaluation methodology.

5.1. Completeness and correctness

In this evaluation step, it was verified that every word should have a tag and that assigned tag should be from the devised POS tagset. At first, we build a utility to assure completeness of word's tags. For tagging, we used slash (/) delimiter between words and tags. It was inspected that no space is inserted between words and tags instead of slash (/). Furthermore, we encountered some tokens whom tags were mistakenly missing. After identifying them we manually assigned tags to the missing words as our utility returned the filename and line number on which issue befell. In the next phase, we checked the correctness of tags. It was assured that tags which have been assigned should be from the devised tagset. For instance, an annotator mistakenly wrote PRH instead of PRF it was highlighted by the utility for correction. At this step of evaluation, it wasn't checked whether the assigned tags are syntactically correct or not. Syntactically correct assignment of tags has been tackled in Section 5.3 and Section 5.4.

5.2. Inter-annotator agreement

To evaluate the quality of annotation, we measured the Inter-Annotator Agreement (IAA) by agreement similarity and Cohen's Kappa (Cohen, 1960). To compute it for all 14 text domains, the reference corpus of 10% was randomly selected from each domain. The statistics of reference corpus according to domains is shown

in Table 6. Two annotators annotated the reference corpus independently. Agreement percentage which is the most basic measure of IAA has been computed by dividing the total number of tags on which the two annotators agree to the total number of tags. Agreement may overestimate the IAA reliability as it doesn't account for chance agreement (Artstein and Poesio, 2008). To overcome this overestimation the Kappa coefficient proposed by Cohen (1960) has been computed for all domains as it recompenses chance agreement. Scale proposed by Landis and Koch (1977) has been used to interpret the Kappa coefficient; values ⩽0 indicates poor agreement, 0.01–0.20 as slight, 0.21–0.40 as fair, 0.41– 0.60 as moderate, 0.61–0.80 as substantial, and 0.81–1.00 leads to almost perfect agreement. The statistics of the IAA similarity and Kappa coefficient for all text domains are shown in Table 6. IAA similarity has been recorded highest 98.2% for the *Business and Economy* domain whereas for the *Classics* genre score was lowest 91.5%. *Classics* domain comprises text which have more ambiguities and entails more words from Hindi which make tagging challenging and more disagreements between annotators. For all domains, average similarity was 95.35% which is more than satisfactory and indicates annotation reliability (Nguyen et al., 2018). Similarly, Kappa coefficient is above 0.90 for all domains which encompass perfect agreement according to Kappa coefficient scale for all genres. Highest Kappa value was reported for *Business and Economy* and *News* domains 0.97 and 0.96 respectively. Kappa value was lowest for *Classics* domain 0.90 due to aforementioned reasons.

Table 6
Statistics of the reference corpus, inter-annotator agreement similarity and Kappa value against each domain.

Sr. #	Domains	#Tokens	#Sentences	Similarity (%)	Kappa
1	Artists	1,225	32	95.6	0.94
2	Business and Economy	762	21	98.2	0.97
3	Comics	880	30	95.4	0.94
4	News	2,272	51	96.8	0.96
5	Sports	762	21	95.1	0.94
6	Short Stories	2,281	81	95.8	0.95
7	Science and Technology	678	20	96.3	0.95
8	Women	917	30	95.3	0.94
9	Novels	1,028	46	95.7	0.95
10	Punjabi Language	2,824	70	95.1	0.94
11	Articles	1,718	54	94.5	0.93
12	Classics	2,115	50	91.5	0.90
13	Emille Dataset	641	20	95.4	0.95
14	Pnb Wikipedia	671	20	94.6	0.93
	Total and Accumulative	19,882	577	95.35	0.94

5.3. Grammar-based consistency evaluation

In the grammar-based consistency evaluation, lexical items that have multiple tags were analyzed and erroneous tags were corrected. POS grammar was extracted from annotated corpus entailing lexical items, corresponding tags assigned to it and the total number of tags in non-increasing order. To maintain consistency throughout the corpus, lexical items with a respective list of tags were analyzed. Syntactically incorrect tags were identified and corrected. Table 7 demonstrates the percentage of words against each POS category before executing grammar-based consistency evaluation and after its execution. For instance, there were almost 10.5% words with two POS tags whereas after applying grammar-based consistency evaluation this percentage was decreased to 7.8%.

Table 8 demonstrates the word statistics against each POS category and the sample POS grammar generated on the evaluated corpus. There are three words which carry seven POS tags and their instances are 1,497 in the corpus. Similarly, 142 words comprise three POS tags with 14,288 instances. Table 8 shows sample POS grammar for word دو *dO* 'two' and CD, VBF and AUXA are correct tags for it. It takes tag CD (cardinal) when its contextual meaning is 'two', VBF (finite verb) when acts as verb and evinces meaning

Table 7
Statistics of the words before and after performing grammar-based consistency evaluation.

	Before		After	
Category	# of Words	Percentage (%)	# of Words	Percentage (%)
Eight POS words	1	0.007	0	0.000
Seven POS words	1	0.007	3	0.021
Six POS words	6	0.042	2	0.014
Five POS words	7	0.049	5	0.035
Four POS words	44	0.311	28	0.198
Three POS words	213	1.508	142	1.005
Two POS words	1,487	10.531	1,115	7.896
One POS words	12,360	87.541	12,824	90.821

Table 8
Word statistics with instances against tags categories and sample POS grammar containing correct POS tags against words with number of unique tags.

Word Statistics Against Each Category			Sample POS Grammar		
Category	#Words	#Instances	Sample Word	POS Tags	Count
Seven POS words	3	1,497	*aE* 'pronoun'	PRP, PDM, VBF, AUXA, AUXT, NNP, NN	7
Six POS words	2	459	*I* 'is'	VBF, AUXT, PRT, PRX, NNP, PRP	6
Five POS words	5	4,610	*tE* 'and'	CC, SCP, SC, PSP, NN	5
Four POS words	28	12,606	*dOvAN* 'give'	VBF, AUXA, OD, NN	4
Three POS words	142	14,288	*dO* 'two'	CD, VBF, AUXA	3
Two POS words	1,115	34,409	*leHandE* 'West'	NNP, JJ	2
One POS words	12,824	987,475	*varHE* 'year'	NN	1

'give' and AUXA (aspectual auxiliary) when appears after verb. Tags such as PSP, SC and AUXT are incorrect for دو *dO* as it doesn't acts as postposition, conjunction and tense auxiliary. Grammar-based consistency evaluation works in semi-automatic way. POS grammar is generated by the utility and implausible tags are manually corrected. A single lexical item is independently analyzed in grammar-based consistency evaluation whereas N-gram based consistency evaluation also considers N words to maintain consistent annotation.

5.4. N-gram based consistency evaluation

In N-gram based consistency evaluation, identical strings that are annotated differently were analyzed. A certain word that occurs more than once can have different tags referred as variation. This variation could be of two types; ambiguity and error. Ambiguity due to lexically possible multiple tags and error as the tagging of a word could be inconsistent athwart comparable occurrences. There will be more chances for the variation to be an error if the context which is composed of words is similar. If in n-gram of words, a word is annotated differently in another occurrence of the identical n-gram, it is referred as variation n-gram. The word that exhibits the variation is stated as the variation nucleus. As if we take insight of two occurrences of the 4-gram variation, (21) and (22). The word *oUh* 'she' is variation nucleus as it shows variation. It is tagged as PRP in (21) which is correct as the personal pronoun 'oUh' is not referring to the noun which is immediately followed by it. Whereas, in (22) it is incorrectly tagged as PDM.

(21)

او	سکول	جاندی	سی
oUh	skUl	jAndI	sI
pron.3.F.Sg	school	go.Pres.3.F.Sg	be.Perf.F.Sg
PRP	NN	VBF	AUXT

'She went to school.'

(22)

او	سکول	جاندی	سی
oUh	skUl	jAndI	sI
pron.3.F.Sg	school	go.Pres.3.F.Sg	be.Perf.F.Sg
PDM	NN	VBF	AUXT

'She went to school.'

For automatic detection of variation n-gram, we have used DECCA[9] tool developed by Dickinson and Meurers (2003). DECCA works independent of the language and the tagset of the corpus. Moreover, no additional resources such as lexica are required. After automatic detection of the variation n-grams, human intervention is required to identify manually whether variation nucleus exhibit an error or it's due to ambiguity. Hence, DECCA works in semi-automatic way. For each variation n-grams, DECCA also provides frequencies of instances. This frequency is used to determine the likelihood of an error as for higher frequency there are less chances of error. Variation n-grams with less frequency are more likely to comprise erroneous tag.

For our corpus, the length of the longest generated variation n-grams was of 33 words. We have evaluated from Bi-grams to 6-grams. For generating n + 1 grams, the DECCA has been executed on n-grams updated corpus. Detailed statistics of the total variation n-grams generated for Bi-gram to 6-gram and correction made for each of them is shown in Table 9. Variation n-grams automatically generated by DECCA have been manually rectified. There were 271 true annotation errors in Bi-grams among 2,131 variation Bi-grams generated. 5-grams were the last n-grams that contained the true tagging error. We also reviewed 6-grams for rectification but didn't find any annotation error in them. So, we decided to perform consistency evaluation till 6-grams. Total 324 corrections are made.

Table 9
Statistics of the variation n-grams generated against Bi-grams to 6-grams and number of corrections made against each.

N-grams	# of Variation n-grams	# of Corrections
Bi-grams	2,131	271
Tri-grams	1,067	46
4-grams	217	6
5-grams	36	1
6-grams	15	0
Total	3,466	324

6. Statistical POS tagging

After corpus evaluation, its final version was divided into training and evaluation set with a standard 80:20 ratio. This distribution was performed randomly across all domains. The evaluation set was equally divided into two sets; test set and development set. The statistics of this distribution are given in Table 10. Training set has been used for taggers training. Development set also known as validation set has been used in our BiLSTM POS tagger. Section 6.2 presents the development and training of our BiLSTM tagger whereas Section 6.3 discusses the training of two statistical POS taggers; TreeTagger and the Stanford tagger. Furthermore, POS results are presented in Section 6.4.

6.1. BiLSTM POS tagger

RNNs are considered ideal for multi-dimensional sequential data processing (Deng, 2014; Du et al., 2020). Our proposed model is based on BiLSTM which is a variant of RNN. As for sculpting sequence labelling, BiLSTMs perform quite proficiently as compared to traditional machine learning algorithms. The architecture of the proposed model is shown in Fig. 3. A BiLSTM consists of two LSTM layers one reads the sequence labelling in the forward direction while on the other hand second reads in backward direction. In doing so, BiLSTM provides additional context to the network for a specific time frame that can efficiently use of past and future features as well. The input sequence of N words $x_1,x_2,\ldots,x_n$ is given as input. For each input vector $x_{1:n}$, a few instances of the vector $h_{1:n}$ are presented in Fig. 3. To perform multi-class classification, a softmax non-linearity function employed at output layer which returns POS tags $t_1,t_2,\ldots,t_n$ for each input sequence of N words x_1, $x_2,\ldots,x_n$. We extended our model with transfer learning as pre-trained word embeddings were also incorporated for a few experiments. Word embeddings also play a vital role in improving sequence tagging performance. Furthermore, character embeddings were utilized which are considered a key signal to improve model performance by learning the morphology of the language. To evaluate the performance of the model, we performed a few experiments with and without word and character embeddings.

For our Shahmukhi BiLSTM POS tagger development, we performed a few experiments with little amendments in our basic model shown in Fig. 3. We started with one BiLSTM layer. To accelerate the tagging results, transfer learning was incorporated. Pre-trained word embeddings have been used to improve the tagging results. Fig. 4 shows the architecture of the bi-directional LSTM based POS tagging model. The training data has shahmukhi words and POS tags in the CoNLL format. The training data is converted in

[9] http://sifnos.sfs.uni-tuebingen.de/decca/.

Table 10
Statistics of the tagged corpus against domains and its division into train, test and development sets.

Sr.#	Domains	#Train Tokens	#Train Sententctes	#Dev Tokens	#Dev Sentences	#Test Tokens	#Test Sentences	#Tokens	# Sentences
1	Artists	9,797	415	1,304	52	887	52	11,988	519
2	Business and Economy	6,727	248	963	31	939	31	8,629	310
3	Comics	4,733	246	523	31	554	30	5,810	307
4	News	13,884	461	1,630	57	1,783	58	17,297	576
5	Sports	4,844	174	626	21	496	22	5,966	217
6	Short Stories	10,259	687	1,309	86	1,326	86	12,894	859
7	Science and Technology	4,372	165	484	21	455	20	5,311	206
8	Women	5,103	234	582	29	730	30	6,415	293
9	Novels	5,524	322	576	40	710	40	6,810	402
10	Punjabi Language	13,570	556	1,729	69	1,759	70	17,058	695
11	Articles	9,043	428	965	52	985	52	11,087	532
12	Classics	9,990	414	1,239	52	1,434	52	12,663	518
13	Emille Dataset	3,124	166	306	21	304	21	3,734	208
14	Pnb Wikipedia	3,507	163	435	21	399	20	4,341	204
	Total	104,477	4,679	12,671	583	12,761	584	130,003	5,846

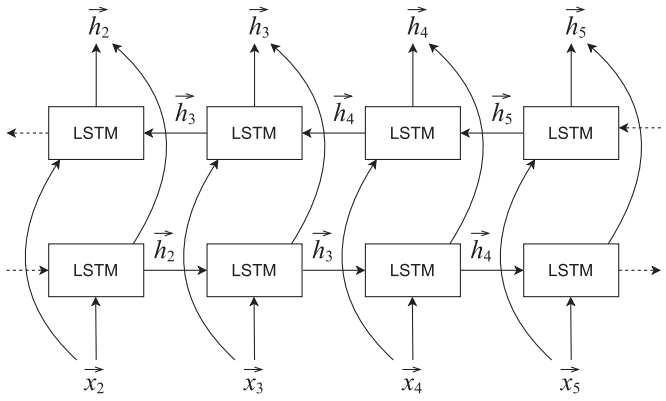

Fig. 3. Bi-directional long-short term memory model for sequence labeling.

the form of vector encodings and fed to the input layer. The vector encodings contain word embeddings and pre-trained word representations. Pre-trained word representations have been used for transfer learning by using context-free and contextualized word representations. The vector representations have been concatenated to achieve a single vector against each token. These vectors are further fed to hidden BiLSTM layers. We have experimented with one and two hidden BiLSTM layers each having 256 units. Hidden layers produce contextual representations which are further given to a dense output layer with *softmax* activation function to perform multi-class classification. The *Adam* optimizer has been used with the learning rate of 0.001. A dropout layer was added with the value of 0.2. The model has been trained for 25 epochs with batch size of 64.

We extended the tagging experiments by including character embeddings in our proposed model. At first, we tried with merely one BiLSTM layer and character embeddings. Later on, we experimented one BiLSTM layer with character embeddings and our pre-trained word embeddings as well. Further, we performed training with two BiLSTM layers. Our Shahmukhi BiLSTM POS tagger performed better with single BiLSTM layer incorporating character embeddings and pre-trained word embeddings. BiLSTM works with fixed length of sentence. Sentence length was set to 181 for all experiments, which was the length of the longest sentence in the test set. Results of experiments are discussed in Section 6.4. Annotation of large datasets is costly in terms of time and expertise. Therefore, we trained BiLSTM based model along with word representations on an unannotated corpus of 14.9 million words. That was all the available Shahmukhi Unicode text on the Internet as per best of our knowledge. However, some other transformer network architectures like BERT and GPT are available but they are mostly used as pretrained models and require a lot of data to train. In future, we will develop a larger corpus for Shahmukhi and will train transformers for POS tagging and other language processing tasks.

6.2. Transfer learning

The neural models are capable of producing state of the art results but they require more data for training. It is very costly to annotate a huge data set for a language, specially a low-resourced language like Shahmukhi. Therefore, transfer learning

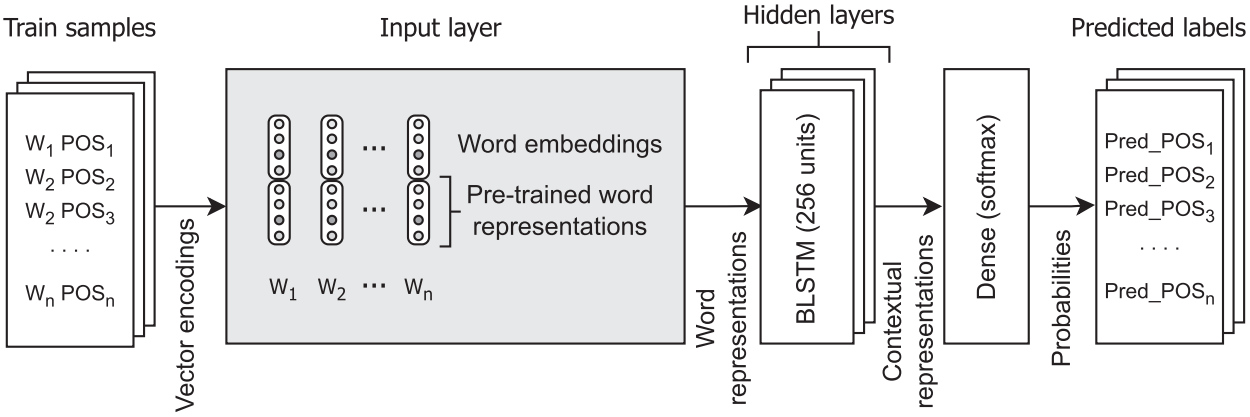

Fig. 4. BiLSTM network based model for Shahmukhi POS tagging.

is a feasible solution to overcome the issue of data sparsity by training word representations on a large corpus without annotating it. It is also helpful to solve the out of vocabulary problem. For the POS tagging, we have trained context-free and contextualized word representations.

6.2.1. Context-free word representations

Context-free word representations produce a feature vector for each word in the corpus. It means a word bears a single meaning beside its multiple semantics in the corpus. However, the context-free word representations are quite capable to learn the syntax and semantics. The well-known word representations are GloVe (Pennington et al., 2014) and Word2Vec (Mikolov et al., 2013) embeddings. These both embeddings are based on different algorithms. The GloVe embeddings are based on word to word co-occurances while Word2Vec uses the co-occurrances of consecutive tokens. In our experiments, we have trained Word2Vec embeddings for transfer learning. Word2Vec embeddings can been obtained by using two techniques, which are skip-gram and continuous bag of words (CBoW). The process of these two techniques is shown by Fig. 5.

The skip-gram model is based on a feed-forward neural network which learns a word from input layer and predicts surrounding words within a given window. The CBoW model, on the other hand, learns the surrounding words and predicts the central word in a window. Context of words can be enhanced by adjusting the window size. Feature vectors are obtained from the hidden layer for both algorithms. Word2Vec embeddings were trained on a Shahmukhi corpus of 14.9 million words which was collected and pre-processed in this study as discussed in Section 3. We obtained the word embeddings of 73 thousand vocabulary size and each word corresponds to a 100-dimensional embedding vector having minimum frequency of five. Shahmukhi being highly agglutinative morphologically-rich language have many word forms from a root word. This productivity enhances Out-of-Vocabulary (OOV) words. An average vector of all the vectors was used agaist each OOV word in our experiments. Word2Vec embeddings have been trained by using gensim library[10].

6.2.2. Contextualized word representations

In a natural language, a word can have multiple meanings associated with it according to the context. The Word2Vec representations are context-free as they do not analyze the word semantics according to context. For the task of POS tagging, the contextualized representations of words can be quite useful. We have trained contextualized word representations (ELMo)[11] (Peters et al., 2018). Embeddings from Language Models (ELMo) is one of the state of the art pre-trained transfer learning models. ELMo representations learn word embeddings from deep bidirectional language model. These representations are character-based which are helpful to learn the morphology of a language. The contextualization is helpful to perform word sense disambiguation and further improves the sequence labeling results. Fig. 6 shows the architecture of ELMo model.

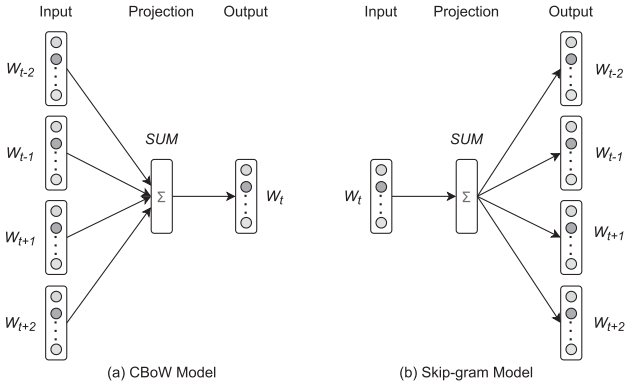

Fig. 5. Architecture of context-free word representations.

ELMo architecture contains three neural network layers which make it faster to train and perform transfer learning. The first layer of the model is the convolutional layer which operates on character level. The convolutional layer is followed by two context dependent bidirectional layers. ELMo word representations are able to produce vectors for OOV words (Ulčar and Robnik-Šikonja, 2019). The contextualization is useful to solve word sense disambiguation and it further improves sequence labeling results. We have trained the ELMo representation by using 14.9 million Shahmukhi corpus. We further analyzed the ELMo representations for Shahmukhi by selecting a tiny corpus which is presented from (23)–(26).

(23)

اوہ	رسوئی	وچ	چاہ	بنا	رہی	سی
oUh	rasOI	vic	cA	bnA	rhI	sI
pron.3.F.Sg	kitchen.F.Sg	case.Loc	tea.F.Sg	make	live.Pres.F.Sg	be.Perf.3.F.Sg
PRP	NN	PSP	NN	VBF	AUXP	AUXT

'She was making tea in the kitchen.'

[10] https://pypi.org/project/gensim.

[11] https://allenai.org/allennlp/software/elmo.

(24)	میرے	دل	وچ	اهناں	دی	بہت	چاہ	ہے
	mErE	dil	vic	OhnAN	dI	boht	cA	hE
	Pron.2.M.Sg	heart.M.Sg	case.Loc	pron.3.Pl	case.Gen	Q	love.F.Sg	be.Pres.Sg
	PRS	NN	PSP	PRP	PSP	Q	NN	VBF

'I have lot of love for them in my heart.'

(25)	اهناں	نے	چاہ	پانی	وی	نہیں	پچھیا
	OhnAN	nE	cA	pAnI	vI	nahIN	pucHyA
	pron.3.Pl	case.Erg	tea.F.Sg	water.M.Sg	also	not	ask.Perf.M.Sg
	PRP	PSP	NN	NN	PRT	NEG	VBF

'They not even asked for tea or water'.

(26)	دل	وچ	سبھ	واسطے	چاہ	تے	محبت	رکھو
	dil	vic	sab	vAstE	cA	tE	muhabbat	rakHO
	heart.M.Sg	case.Loc	all	for	love.F.Sg	and	love.F.Sg	keep.Pres.Sg
	NN	PSP	Q	PSP	NN	CC	NN	VBF

'Keep the love for everyone in you heart.'

The small corpus has been selected for the semantic analysis of a Shahmukhi word چاہ '*cA*'. This word has the senses of 'tea' and 'love' in the language. Both meanings are quite different from each other. Diacritic marks are optional in Shahmukhi script as it uses Urdu script for writing therefore, we have not used diacritics in these four sentences. The first sentence (23) is about making tea

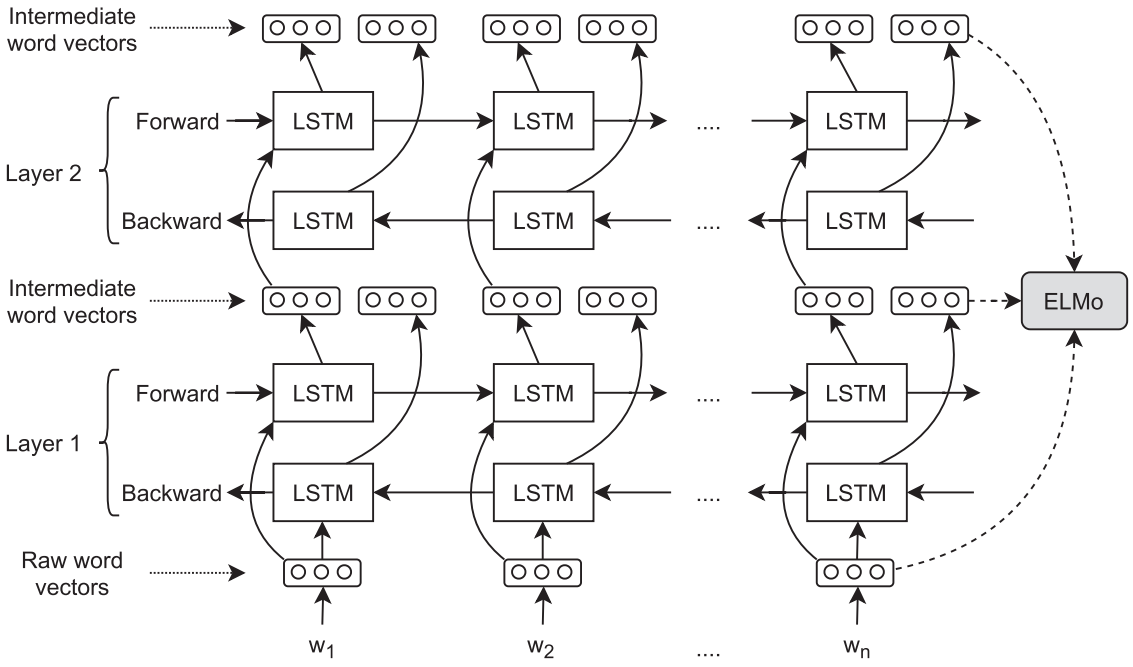

Fig. 6. Architecture of contextualized word representations.

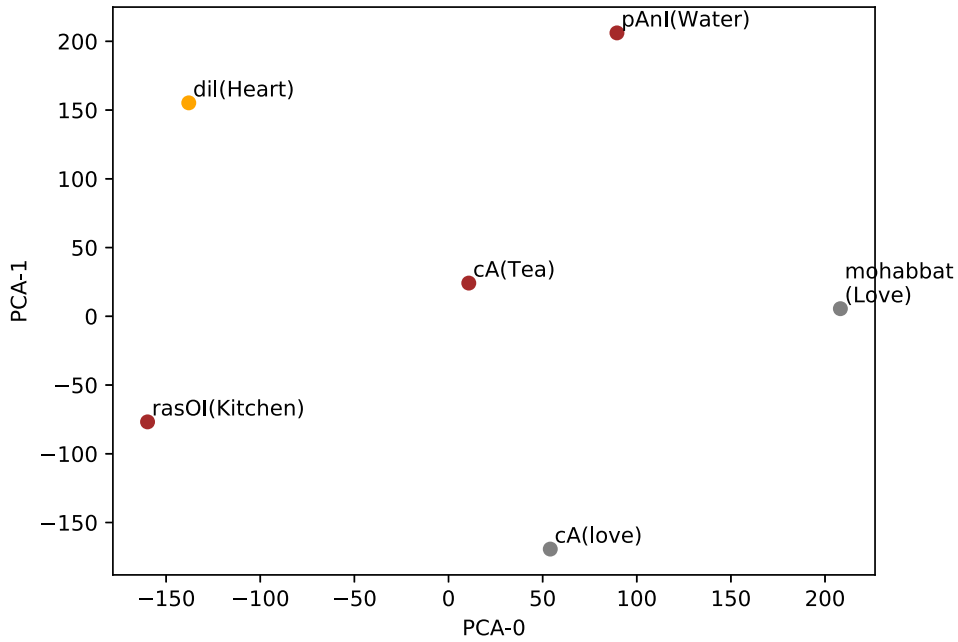

Fig. 7. Feature vector representations of words based on their contexts in the selected corpus of four sentences.

Table 11
Cosine similarity of word vectors achieved from ELMo and Word2Vec representations for the sample corpus.

	ELMo		Word2Vec	
Word vectors	*cA* 'Tea'	*cA* 'Love'	*cA* 'Tea'	*cA* 'Love'
rasOI 'Kitchen'	**0.74622944**	0.51053325	0.51078500	0.51078500
pAnI 'Water'	**0.49145018**	0.48223738	0.52641760	0.52641760
dil 'Heart'	0.31365096	**0.51430908**	0.20029573	0.20029573
mohabbat 'Love'	0.35048401	**0.77118605**	0.12898123	0.12898123

in the kitchen. The second sentence is about the love for someone in heart. The third sentence again discusses the tea and water which both are drinkable liquids. The forth sentence tells that one should have love for everyone in heart. We obtained ELMo vectors for the small corpus and used a 2-dimensional plane to plot the prominent words as shown in Fig. 7.

The ELMo representations have been trained with a vocabulary size of 73 thousands and feature vectors of 128 dimensions. Principle Component Analysis (PCA) has been used to perform dimensionallity reduction. It is clearly shown in Fig. 7 that the word چاہ '*cA*' has two different meanings 'tea' and 'love'. The graph shows same colors for semantically similar words. The words *cA* 'tea', *pAnI* 'water' and *rasOI* 'kitchen' are represented with brown dots. The words *cA* 'love' and *mohabbat* 'love' are represented with gray color. Similarly, the word *dil* 'heart' is shown with orange color. Semantically similar words make clusters in the 2-dimensional plane. The word *cA* 'tea' is near to the words *rasOI* 'kitchen' and *pAnI* 'water'. Similarly, the word *cA* 'love' is near to the word *mohabbat* 'love'.

We further computed cosine similarities for four words, *rasOI* 'kitchen', *pAnI* 'water', *dil* 'heart' and *mohabbat* 'love' with two representations of *cA* as shown by Table 11. The similarities have been computed for both types of word representations, contextualized ELMo and context-free Word2Vec. The word vectors from ELMo for the word *cA* clearly shows two meanings. The vector for *cA* 'tea' has higher similarities with *rasOI* 'kitchen' and *pAnI* 'water' as compared to *dil* 'heart' and *mohabbat* 'love'. Similarly, the word *cA* 'love' has higher similarity with *dil* 'heart' and *mohabbat* 'love' as compared to *rasOI* 'kitchen' and *pAnI* 'water'. On the other hand, the Word2Vec representation has static word vectors for words. Both word vectors of *cA* have same cosine similarities with four other words as shown in Table 11. The Word2Vec representation of *cA* has a single meaning of 'tea' as it shows higher similarities with *rasOI* 'kitchen' and *pAnI* 'water' and lower similarities with *dil* 'heart' and *mohabbat* 'love'. The analysis of ELMo and Word2Vec representations shows that Word2Vec learns context-free and static vectors while ELMo representations produce contextualized word vectors. We have performed the POS tagging experiments by employing Word2Vec and ELMo embeddings and the results are presented in Section 6.4.

6.3. Other statistical POS taggers

In this study, we selected two well-known and language-independent statistical POS taggers; TreeTagger[12] and Stanford tagger[13]. (Schmid, 1994.) proposed the TreeTagger which uses decision trees for storing transition probabilities then the trained decision tree is used to assign the highest probable tags. On Penn Treebank, TreeTagger performed with an accuracy of 96.36%. The Stanford POS tagger uses a bidirectional version of maximum entopy markov model called cyclic dependency network as its core algorithm (Toutanova et al., 2003). On Penn Treebank, it performed with an accuracy of 97.24%. It also has been trained for numerous languages

[12] https://www.cis.uni-muenchen.de/schmid/tools/TreeTagger/.
[13] https://nlp.stanford.edu/software/tagger.shtml

Table 12
Our BiLSTM POS tagger results for all experiments conducted with different features. The BiLSTM tagger outclassed with features that are in bold text.

Sr.#	BiLSTM Tagger + Features	Precision	Recall	F-score	Accuracy(%)
1	1 BiLSTM	92.61	92.63	92.56	92.63
2	1 BiLSTM + Ch_emb	94.13	92.97	92.98	92.97
3	1 BiLSTM + W2V	94.36	94.36	94.33	94.36
4	2 BiLSTM + W2V	94.43	94.41	94.38	94.41
5	**1 BiLSTM + W2V + Ch_emb**	**95.57**	**95.51**	**95.51**	**95.51**
6	2 BiLSTM + W2V + Ch-emb	94.87	94.86	94.82	94.86
7	**1 BiLSTM + W2V + ELMo**	**96.12**	**96.12**	**96.11**	**96.12**
8	2 BiLSTM + W2V + ELMo	96.05	96.02	0.96.02	96.02

and achieved decent tagging accuracy; Chinese (94.13%) (Yu and Chen, 2012), Urdu (96.4%) (Naseem et al., 2017), Arabic (96.26%), German (96.9%) and Filipino (96.15%) (Go and Nocon, 2017).

TreeTagger requires labeled data and a lexicon. For training, corpus was developed but in lexicon we used spurious lemmas as no lemmatizer was available. All noun, noun modifiers, verb, adverb and auxiliary verbs tags were marked as open class tags. A decision tree of 113 number of nodes with 31 maximum path length was generated by TreeTagger during training. After building the training model, we tested the created model on our test set.

Further, we trained Stanford tagger. The Stanford tagger requires only annotated data for training. Stanford tagger's general properties file was created and some features were updated; iterations were set to 100 and generic feature extractor was used. Same open class tags which were specified for TreeTagger model were used for Stanford Tagger training. Stanford tagger comparatively took more time for training but performed better than TreeTagger. Performance results of both taggers in comparison with our BiLSTM POS tagger are presented in Section 6.4.

6.4. Results and discussion

For evaluating POS taggers, test corpus of 12,761 tokens has been used. Test set detailed statistics are shown in Table 10. To calculate performance of the taggers, precision, recall, f-score and accuracy were calculated by using following equations.

$$Precision = \frac{No.\ of\ correct\ tags\ in\ candidate\ set}{Total\ tags\ in\ candidate\ set} \quad (1)$$

$$Recall = \frac{No.\ of\ correct\ tags\ in\ candidate\ set}{Total\ tags\ in\ reference\ set} \quad (2)$$

$$F_1 Score = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \quad (3)$$

$$Accuracy = \left(\frac{No.\ of\ correct\ tags}{Total\ no.\ of\ tags}\right) \cdot 100 \quad (4)$$

Table 12 evinces the results of all experiments that were performed with the BiLSTM POS tagger. We started the BiLSTM POS tagger training by merely single hidden layer without character embeddings and transfer learning. It performed with an f-score of 92.56 and 92.63% accuracy. The character embeddings improved the tagging results and gave an increase of 0.42 in the f-score. However, using word embeddings in the training gave significant improvements in the results. BiLSTM tagger using two hidden LSTM layers produced POS results with an f-score of 94.38 and accuracy of 94.41%. Word2Vec embeddings along with character embeddings further improved the results to an f-score of 95.51 and accuracy of 95.51%. ELMo word representations along with Word2Vec embeddings outperformed all other models with an f-score of 96.11 and 96.12% accuracy. However, the ELMo and Word2Vec with two hidden LSTM layers reduced the f-score with the factor of 0.1. The POS tagging results are quite promising for a low-resourced and morphologically-rich language Shahmukhi.

Table 13 presents our BiLSTM POS tagger results in comparison with two other statistical taggers; TreeTagger and the Stanford tagger. The TreeTagger performed with 85.17 f-score and 85.77% accuracy. The Stanford tagger performed better than TreeTagger with 94.36 f-score and 94.43% accuracy. Results were comparatively less satisfying for TreeTagger due to spurious lemmas that we used in training. In terms of training time, BiLSTM tagger leads to an expensive model as it requires more training time in comparison with the TreeTagger and the Stanford tagger. Among all three taggers, the BiLSTM POS tagger outperformed with an f-score of 96.11 and an accuracy of 96.12%.

We further interpreted the BiLSTM POS tagger performance domain and tag wise. Domain wise results are demonstrated in Table 14 whereas individual POS tags results are presented in Table 15. These results have been generated by comparing predicted POS tags with gold POS tags. By taking accuracy metrics into account, *News* domain performed best with an accuracy of 97.53%. Whereas, f-score was highest for *Novels* domain with an accuracy of 97.46% which is quite high. Among all 14 domains, *Emille Dataset* domain recorded lowest f-score 88.81 and accuracy 94.73%. Indeed, these results for *Emille Dataset* also indicate good performance.

In Table 15 individual tags results are presented. Support against all tags is shown with precision, recall and f-score. Support column refers to the number of occurrences of a tag in the test corpus. F-scores are higher for tags; APNA, FF, FR, PRF, SCP, PU, SYM and VALA indicating perfect precision and recall. The tagger has given performance against these tags because words under these tags don't face lexical ambiguity. For example, FR with support of six, reported 1.00 f-score. The words having this tag usually are not marked with any other tags. The word *DHAI* 'two and half' in (1) is annotated to have FR tag. Due to morphological richness of Shahmukhi, it is challenging for taggers to disambiguate between some pairs of tags such as between NN and NNP, PDM and PRP, NN and JJ, SC and CC, VBF and AUXA etc. For instance, as Shahmukhi doesn't rule out any distributional distinction between common noun (NN) and proper noun (NNP) at syntactic level. Such as the word بخش *baxS* 'forgive' is tagged as NNP when it refers to the name of a person and is tagged as NN when it refers to a common noun munificence. However, due to cross-linguistic differences, these tags are vital to be kept separate. NN with highest support of 3,090 reported an f-score of 95.76 whereas NNP

Table 13
Statistical POS taggers results.

Sr. #	Tagger	Precision	Recall	F-score	Accuracy (%)
1	**1 BiLSTM + W2V + ELMo**	**96.12**	**96.12**	**96.11**	**96.12**
2	TreeTagger	86.30	85.78	85.71	85.77
3	Stanford Tagger	94.42	94.44	0.9436	94.43

Table 14
Our BiLSTM POS tagger domain wise results.

Sr.#	Domain	Precision	Recall	F-score	Accuracy(%)
1	Artists	97.17	95.79	96.35	95.71
2	Business and Economy	96.15	96.3	96.06	95.95
3	Comics	93.65	94.00	93.62	95.66
4	News	98.16	97.71	97.84	97.53
5	Sports	95.62	93.22	94.27	95.16
6	Short Stories	90.51	90.74	90.41	96.68
7	Science and Technology	93.72	93.85	93.45	96.92
8	Women	95.86	97.19	96.18	96.84
9	Novels	97.43	96.37	96.83	97.46
10	Punjabi Language	93.44	92.02	92.14	95.39
11	Articles	91.08	89.29	89.73	96.64
12	Classics	94.46	95.13	94.66	94.35
13	Emillie Dataset	89.31	89.98	88.81	94.73
14	Pnb Wikipedia	95.15	94.69	93.98	95.73

Table 15
Our BiLSTM POS tagger results for individual tags.

Sr.#	POS tags	Precision	Recall	F-score	Support
1	APNA	1.00	1.00	1.00	69
2	AUXA	96.68	96.18	96.43	393
3	AUXM	98.25	98.25	98.25	57
4	AUXP	1.00	96.55	98.25	58
5	AUXT	99.09	98.64	98.86	441
6	CC	86.74	91.32	88.97	265
7	CD	96.46	98.79	97.61	248
8	FF	1.00	1.00	1.00	7
9	FR	1.00	1.00	1.00	6
10	INJ	88.46	85.19	86.79	27
11	JJ	91.54	91.18	91.36	748
12	NEG	1.00	99.29	99.64	141
13	NN	95.41	96.12	95.76	3090
14	NNP	92.47	92.27	92.37	932
15	OD	94.2	95.59	94.89	68
16	PDM	88.73	94.76	91.65	191
17	PRD	75.00	75.00	75.00	8
18	PRE	1.00	80.00	88.89	5
19	PRF	1.00	1.00	1.00	11
20	PRP	97.8	94.68	96.22	470
21	PRR	97.37	96.52	96.94	115
22	PRS	1.00	98.92	99.46	93
23	PRT	97.32	96.46	96.89	226
24	PRX	75.00	60.00	66.67	5
25	PSP	98.84	98.94	98.89	1887
26	PU	1.00	99.81	99.9	1045
27	Q	93.65	89.39	91.47	132
28	QM	0.00	0.00	0.00	2
29	RB	99.15	91.41	95.12	128
30	SC	92.19	88.06	90.08	201
31	SCK	97.59	98.78	98.18	82
32	SCP	1.00	1.00	1.00	32
33	SYM	97.7	1.00	98.84	85
34	VALA	1.00	1.00	1.00	54
35	VBF	96.53	96.53	96.53	1181
36	VBI	95.37	95.74	95.55	258

f-score was 92.37 with 932 support. F-score of SC was 90.08 with 201 support whereas CC tag f-score was 88.97 with 265 support. F-score of SC (subordinate conjunction) and CC (coordinate conjunction) tags are effected due to common words ambiguity, consequently, these tags are used interchangeably based on context of a word, such as *tE* 'and/then' in (8) is CC and in (1) it is SC.

F-scores of pronouns, verbs, auxiliaries verbs and particle categories are highly affected due to lexical ambiguities. For example, the word ے\ *aE* takes the tags NN, NNP, VBF, AUXA, AUXT, PRP and PDM as well. Such as in 5 its tag is AUXT, in 3 it is PDM, in 13 VBF and in 14 it is PRP. Tagging accuracy of such words could be increased by having diverse use of lexical items and variety of syntactic patterns. Support of a tag in train set is also important to elevate scores. Lowest f-scores are reported for QM tag which is due to less number of occurrences in training and test sets. The occurrences of QM were five and two in train and test set respectively. Word *dUgnI* 'double' is an example of QM tag in (13). PRD and PRX results were much better than QM but are recorded less f-score comparatively as for PRD and PRX it was 75.00 and 66.67 respectively. F-score of VBF was 96.53 with 1181 support. F-scores and accuracies of all other POS tags are quite satisfying.

7. Conclusion

We present the development of a Shahmukhi POS tagged corpus and a BiLSTM POS tagger. A Shahmukhi corpus of 14.9 million words was collected and prepared. After preprocessing, the corpus was divided into 14 text domains based on the contents. A balanced genre based corpus of 0.13 million words was extracted for annotation. For the corpus annotation, a POS tagset was devised and annotation guidelines have been presented. It was analyzed that grammatical similarities exist between Urdu and Shahmukhi such as morphological richness, case markers, complex predicate, lexical ambiguity and copula constructions. Therefore, a Shahmukhi POS tagset has been devised by studying the applicability of an Urdu POS tagset; the CLE Urdu POS tagset. The CLE Urdu POS tagset covers morpho-syntactic categories which are applicable on Shahmukhi. However, all tags remain same for Shahmukhi with an additional tag pronominal suffixes under main category of pronoun. Conclusively, a Shahmukhi POS tagset of 36 tags has been used for tagging. Further, a multi-step corpus evaluation process was employed. At first, completeness and correctness of tags was assured which was followed by a manual review of the annotated corpus. Corpus evaluation was extended with two consistency evaluation steps; grammar-based and n-gram based consistency evaluations. The inter-annotator agreement also has been computed for reference corpus which recorded average IAA similarity of 95.35% and average Kappa value of 0.94 leading to perfect agreement for all 14 domains. A bi-directional long-short term memory based POS tagger has been developed. For developing the BiLSTM POS tagger, a few experiments were performed with different features. Furthermore, two other statistical taggers were trained; TreeTagger and Stanford tagger. Our BiLSTM POS tagger outperformed with an f-score of 96.11 and an accuracy of 96.12%. In the future, our work will serve as a milestone for the development of Shahmukhi digitally and will also enable future computational research for Shahmukhi like syntactic parsing, text summarization, text to speech systems and information extraction

etc. Overall, this work presents five main contributions for Shahmukhi language which include, a POS tagged corpus, POS tagset and annotation guidelines, a cleaned text corpus of 14.9 million words, a neural POS tagger, and word embeddings.

Declaration of Competing Interest

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

Acknowledgement

We are thankful to Qatar National Library (QNL) and Hamad Bin Khalifa University (HBKU) for supporting the publication charges of this paper.

References

Abbas, Q., 2014. Semi-semantic part of speech annotation and evaluation. In: Proceedings of LAW VIII-The 8th Linguistic Annotation Workshop, pp. 75–81.
Ahmad, M.T., Malik, M.K., Shahzad, K., Aslam, F., Iqbal, A., Nawaz, Z., Bukhari, F., 2020. Named entity recognition and classification for Punjabi Shahmukhi. ACM Trans. Asian Low-Resource Lang. Informat. Process. (TALLIP) 19, 1–13.
Ahmed, T., 2007. Ablative, sociative and instrument markers in Urdu, Punjabi and Sindhi. University of Konstanz1-Stuttgart.
Ahmed, T., Urooj, S., Hussain, S., Mustafa, A., Parveen, R., Adeeba, F., Hautli, A., Butt, M., 2015. The CLE Urdu POS tagset. In: LREC 2014, Ninth International Conference on Language Resources and Evaluation, pp. 2920–2925.
Akhter, N., Mahmood, M.A., Nadeem, M.T., 2019. Desarrollo de nombres en Punjabi y relaciones léxico-semánticas. Dilemas Contemporáneos: Educación, Política y Valores 6.
Anwar, W., Wang, X., Li, L., Wang, X.L., 2007. A statistical based part of speech tagger for Urdu language. In: 2007 International Conference on Machine Learning and Cybernetics, IEEE. pp. 3418–3424.
Arslan, M.F., Mehmood, M.A., Hayat, S., 2019. Estudio basado en corpus sobre el perfil de vocabulario del lenguaje Shahmukhi Punjabi. Dilemas Contemporáneos: Educación, Política y Valores 6.
Artstein, R., Poesio, M., 2008. Inter-coder agreement for computational linguistics. Comput. Linguist. 34, 555–596.
Bhurgari, A., 2007. Enabling Pakistani languages through Unicode, published at http://download.microsoft.com/download/1/4/2/142aef9f-1a74-4a24-b1f4-782d48d41a6d.PakLang.pdf.
Butt, M., 1995. The structure of complex predicates in Urdu. Center for the Study of Language (CSLI).
Butt, M., 2007. The role of pronominal suffixes in Punjabi. Architecture, Rules, Preferences, 341–368.
Butt, M., King, T.H., 2003. Case systems: Beyond structural distinctions.
Charniak, E., Hendrickson, C., Jacobson, N., Perkowitz, M., 1993. Equations for part-of-speech tagging. In: AAAI, pp. 784–789.
Cohen, J., 1960. A coefficient of agreement for nominal scales. Educ. Psychol. Measur. 20, 37–46.
Deng, L., 2014. A tutorial survey of architectures, algorithms, and applications for deep learning. APSIPA Trans. Signal Informat. Process. 3.
Dickinson, M., Meurers, W.D., 2003. Detecting errors in part-of-speech annotation. In: Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL-03), Budapest, Hungary. pp. 107–114. URL: http://ling.osu.edu/dm/papers/dickinson-meurers-03.html.
Dua, M., Aggarwal, R., Kadyan, V., Dua, S., 2012. Punjabi automatic speech recognition using HTK. Int. J. Comput. Sci. Issues (IJCSI) 9, 359.
Du, J., Vong, C.M., Chen, C.P., 2020. Novel efficient RNN and LSTM-like architectures: Recurrent and gated broad learning systems and their applications for text classification. IEEE Trans. Cybernet.
Ehsan, T., 2022. Statistical Parser for Urdu. University of Engineering and Technology, Lahore, Pakistan. Ph.D. thesis.
Ehsan, T., Butt, M., 2020. Dependency parsing for Urdu: Resources, conversions and learning. In: Proceedings of the 12th Language Resources and Evaluation Conference, pp. 5202–5207.
Ehsan, T., Hussain, S., 2019. Analysis of experiments on statistical and neural parsing for a morphologically rich and free word order language Urdu. IEEE Access 7, 161776–161793.
Ehsan, T., Hussain, S., 2020. Development and evaluation of an Urdu treebank (CLE-UTB) and a statistical parser. Language Resourc. Eval., 1–40
Ehsan, T., Khalid, J., Ambreen, S., Mustafa, A., Hussain, S., 2022. Improving Phrase Chunking by using Contextualized Word Embeddings for a Morphologically Rich Language. Arabian J. Sci. Eng. 47, 9781–9799.
Giménez, J., Marquez, L., 2004. Fast and accurate part-of-speech tagging: The SVM approach revisited. Recent Adv. Natural Language Process. III, 153–162.
Go, M.P., Nocon, N., 2017. Using Stanford part-of-speech tagger for the morphologically-rich Filipino language. In: Proceedings of the 31st Pacific Asia Conference on Language, Information and Computation, pp. 81–88.
Gupta, V., Kaur, N., 2016. A novel hybrid text summarization system for Punjabi text. Cognitive Comput. 8, 261–277.
Gupta, V., Lehal, G.S., 2011. Punjabi language stemmer for nouns and proper names. In: Proceedings of the 2nd Workshop on South Southeast Asian Natural Language Processing (WSSANLP), pp. 35–39.
Hardie, A., 2003. Developing a tagset for automated part-of-speech tagging in Urdu. In: Corpus Linguistics 2003.
Hasan, E., Iqbal, M.M., Azeemi, Q.R., Javeed, A., 2015. An online Punjabi Shahmukhi lexical resource. Sci. Int (Lahore) 27, 2529–2535.
Hashmi, M.A., Mahmood, M.A., Mahmood, M.I., 2019a. Analysis of lecxico-semantic relations of Punjabi Shahmukhi nouns: A corpus based study. Int. J. English Linguist. 9.
Hashmi, M.A., Mahmood, M.A., Mahmood, M.I., 2019b. A corpus based study of developing lexicosemantic relations among the verbs of Punjabi Shahmukhi. J. Organ. Behav. Res. 4, 162–170.
Humayoun, M., Ranta, A., 2010. Developing Punjabi morphology, corpus and lexicon. In: Proceedings of the 24th Pacific Asia Conference on Language, Information and Computation, pp. 163–172.
Jawaid, B., Kamran, A., Bojar, O., 2014. A tagged corpus and a tagger for Urdu. In: LREC, pp. 2938–2943.
Kamran Malik, M., Ahmed, T., Sulger, S., Bögel, T., Gulzar, A., Raza, G., Hussain, S., Butt, M., 2010. Transliterating Urdu for a broad-coverage Urdu/Hindi LFG grammar. In: LREC 2010, Seventh International Conference on Language Resources and Evaluation, pp. 2921–2927.
Kaur, M., Aggerwal, M., Sharma, S.K., 2014. Improving Punjabi part of speech tagger by using reduced tag set. Int. J. Comput. Appl. Informat. Technol. 7, 142.
Kaur, A., Singh, P., Kaur, K., 2017. Punjabi dialects conversion system for Majhi, Malwai and Doabi dialects. In: Proceedings of the 8th International Conference on Computer Modeling and Simulation, pp. 125–128.
Khan, T.A., 2009. Spatial expressions and case in South Asian languages. Ph.D. thesis.
Khan, W., Daud, A., Khan, K., Nasir, J.A., Basheri, M., Aljohani, N., Alotaibi, F.S., 2019. Part of speech tagging in Urdu: Comparison of machine and deep learning approaches. IEEE Access 7, 38918–38936.
Khan, W., Daud, A., Nasir, J.A., Amjad, T., Arafat, S., Aljohani, N., Alotaibi, F.S., 2019. Urdu part of speech tagging using conditional random fields. Language Resourc. Eval. 53, 331–362.
Kiperwasser, E., Goldberg, Y., 2016. Simple and accurate dependency parsing using bidirectional LSTM feature representations. Trans. Assoc. Comput. Linguist. 4, 313–327.
Kumar, D., Josan, G.S., 2012. Developing a tagset for machine learning based POS tagging in Punjabi. Int. J. Appl. Res. Informat. Technol. Comput. 3, 132–143.
Kumar, D., Josan, G., 2016. Prediction of part of speech tags for Punjabi using support vector machines. Int. Arab J. Informat. Technol. (IAJIT) 13.
Lafferty, J., McCallum, A., Pereira, F.C., 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.
Landis, J.R., Koch, G.G., 1977. The measurement of observer agreement for categorical data. Biometrics, 159–174.
Lehal, G.S., 2009. A Gurmukhi to Shahmukhi transliteration system. In: Proceedings of ICON-2009: 7th International Conference on Natural Language Processing, pp. 167–173.
Malik, M.G.A., 2006. Punjabi machine transliteration. In: 21st international Conference on Computational Linguistics (COLING) and the 44th Annual Meeting of the ACL, pp. 1137–1144.
Mateen, A., Malik, M.K., Nawaz, Z., Danish, H., Siddiqui, M.H., Abbas, Q., 2017. A hybrid stemmer of Punjabi Shahmukhi script. Int. J. Comput. Sci. Netw. Secur. 17, 90–97.
Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J., 2013. Distributed representations of words and phrases and their compositionality. Adv. Neural Informat. Process. Syst., 3111–3119
Mittal, S., Sethi, N.S., Sharma, S.K., 2014. Part of speech tagging of Punjabi language using N gram model. Int. J. Comput. Appl. 100.
Mukund, S., Srihari, R., Peterson, E., 2010. An information-extraction system for Urdu—a resource-poor language. ACM Trans. Asian Language Informat. Process. (TALIP) 9, 1–43.
Naseem, A., Anwar, M., Ahmed, S., Jan, A., Malik, A.K., 2017. Reusing Stanford POS tagger for tagging Urdu sentences. In: 2017 13th International Conference on Emerging Technologies (ICET), IEEE. pp. 1–6.
Nasim, Z., Abidi, S., Haider, S., 2020. Modeling POS tagging for the Urdu language. In: 2020 International Conference on Emerging Trends in Smart Technologies (ICETST), IEEE. pp. 1–6.
Nguyen, Q.T., Miyao, Y., Le, H.T., Nguyen, N.T., 2018. Ensuring annotation consistency and accuracy for Vietnamese treebank. Language Resourc. Eval. 52, 269–315.
Nivre, J., Hall, J., Nilsson, J., Chanev, A., Eryigit, G., Kübler, S., Marinov, S., Marsi, E., 2007. Maltparser: A language-independent system for data-driven dependency parsing. Natural Language Eng. 13, 95–135.
PBS, 2017. Population by mother tongue: Pakistan Bureau of Statistics. URL: http://www.pbs.gov.pk/content/population-mother-tongue.
Pennington, J., Socher, R., Manning, C.D., 2014. GloVe: Global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543.
Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L., 2018. Deep contextualized word representations. arXiv preprint arXiv:1802.05365.

Ratnaparkhi, A., 1996. A maximum entropy model for part-of-speech tagging. In: Conference on Empirical Methods in Natural Language Processing.
Saini, T.S., Lehal, G.S., 2011. Word disambiguation in Shahmukhi to Gurmukhi transliteration. In: Proceedings of the 9th Workshop on Asian Language Resources, pp. 79–87.
Saini, T.S., Lehal, G.S., Kalra, V.S., 2008. Shahmukhi to Gurmukhi transliteration system. In: Coling 2008: Companion volume: Demonstrations, pp. 177–180.
Sajjad, H., 2007. Statistical part of speech tagger for Urdu. Unpublished MS Thesis, National University of Computer and Emerging Sciences, Lahore, Pakistan.
Sajjad, H., Schmid, H., 2009. Tagging Urdu text with parts of speech: A tagger comparison. In: Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009), pp. 692–700.
Schmid, H., 1994. Probabilistic part-of-speech tagging using decision trees. In: International Conference on New Methods in Language Processing, 1994.
Schmid, H., 2013. Probabilistic part-of-speech tagging using decision trees. in: New Methods in Language Processing, p. 154.
Sharma, S.K., 2016. Assigning the correct word class to Punjabi unknown words using CRF. Int. J. Comput. Appl. 975, 8887.
Sharma, S.K., Lehal, G.S., 2011. Using hidden markov model to improve the accuracy of Punjabi POS tagger. In: 2011 IEEE International Conference on Computer Science and Automation Engineering, IEEE. pp. 697–701.
Simons, G.F., Fennig, C.D., 2017. Ethnologue: languages of Asia. Sil International Dallas.
Singh, G., 2008. Development of Punjabi grammar checker. Phd. Dissertation.
Singh, P., Lehal, G.S., 2006. Text-to-speech synthesis system for Punjabi language. In: Proceedings of International Conference on Multidisciplinary Information Sciences and Technologies, Merida, Spain.
Singh, S.K., Sachan, M.K., 2019. Grt: Gurmukhi to Roman transliteration system using character mapping and handcrafted rules. Int. J. Innovat. Technol. Expl. Eng. 8, 2758–2763.
TDIL, 2014. punjabipos.learnpunjabi.org/tagset.aspx. URL: http://punjabipos.learnpunjabi.org/TagSet.aspx. (Accessed on 09/30/2020).
Toutanova, K., Klein, D., Manning, C.D., Singer, Y., 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. in: Proceedings of the 2003 conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-volume 1, Association for Computational Linguistics. pp. 173–180.
Ulčar, M., Robnik-Šikonja, M., 2019. High quality ELMo embeddings for seven less-resourced languages. arXiv preprint arXiv:1911.10049.
Urooj, S., Shams, S., Hussain, S., Adeeba, F., 2014. Sense tagged CLE Urdu digest corpus. In: Proc. Conf. on Language and Technology, Karachi.
Walia, H., Rana, A., Kansal, V., 2019. Case based interpretation model for word sense disambiguation in Gurmukhi. in: 2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence), IEEE. pp. 359–364.
Yu, C.H., Chen, H.H., 2012. Chinese web scale linguistic datasets and toolkit. In: Proceedings of COLING 2012: Demonstration Papers, pp. 501–508.