Arabian Journal for Science and Engineering (2022) 47:9781–9799
https://doi.org/10.1007/s13369-021-06343-7

RESEARCH ARTICLE-COMPUTER ENGINEERING AND COMPUTER SCIENCE

Improving Phrase Chunking by using Contextualized Word Embeddings for a Morphologically Rich Language

Toqeer Ehsan[1] · Javairia Khalid[2] · Saadia Ambreen[2] · Asad Mustafa[2] · Sarmad Hussain[2]

Received: 6 September 2021 / Accepted: 7 October 2021 / Published online: 2 December 2021

Abstract
Phrase chunking is an important task in various natural language processing (NLP) applications. This paper presents a neural phrase chunking for Urdu by training contextualized word representations. This work also produces an annotated corpus. The annotation has been performed by using IOB (inside-outside-begin) labels. Comprehensive guidelines have been developed for four phrases which are noun phrase (NP), verb phrase (VP), post-positional phrase (PP) and prepositional phrase (PRP). The annotated text has been evaluated for completeness and correctness automatically. Inter-annotator agreement has been calculated for ten percent reference corpus. A neural chunker has been developed and trained on the annotated corpus. The chunker is based on long–short- term memory networks. Transfer learning has been employed to improve the chunking results. For that purpose, context-free (Word2Vec) and contextualized (ELMo) word representations have been trained. The chunker performed with an f-score of 94.9 when trained by using third layer of ELMo embeddings.

Keywords BiLSTM · ELMo · Urdu · Chunking · Shallow Parsing

1 Introduction

Urdu is an Indo-Aryan language which is written in a version of Arabic script. It is mainly spoken in South Asia and has more than 160 million speakers all over the world [1]. Urdu has several prominent linguistic properties that are morphological richness [2,3], complex predicate structure [4,5], case system [6,7] and flexible word order [8]. Due to these features, it is quite challenging to achieve higher performance for the tasks of language processing. Several computational resources have been built in the past decade which include text corpora, morphological analysis, word segmentation, part of speech tagging and syntactic parsing. However, it is still under-resourced and there is a need to built the essential computational resources in comparison with the resource-rich languages. The phrase chunking (shallow parsing) is a process to segment sentences into sequences of constituents or chunks, i.e., the sequences of adjacent words are grouped on the basis of linguistic properties. Chunk parsing is also contemplated as an intermediate step to the full parsing. Chunking is introduced as a response to the difficulties of full parsing and presents text efficiently without indulging in deeper analysis. The phrase chunking is used in different natural language processing tasks including Named Entity Recognition (NER), Terminology Discovery [9] and Text Mining [10].

Chunks can be represented using either trees or tags. The most widespread representation of chunks is by using the IOB tags. In the IOB tagging scheme, each chunk is represented using three special tags, I (inside), O (outside) or B (begin). A token is tagged with tag B if it marks the beginning of a chunk. The subsequent tokens which are inside of a chunk

Toqeer Ehsan and Javairia Khalid These authors contributed equally to this work.

✉ Toqeer Ehsan
toqeer.ehsan@uog.edu.pk

Javairia Khalid
javairia.khalid@kics.edu.pk

Saadia Ambreen
saadia.ambreen@kics.edu.pk

Asad Mustafa
asad.mustafa@kics.edu.pk

Sarmad Hussain
sarmad.hussain@kics.edu.pk

[1] Department of Computer Science, University of Gujrat, Gujrat 50700, Pakistan

[2] Center for Language Engineering (CLE), Al-Khawarizmi Institute of Computer Science (KICS), University of Engineering and Technology (UET), Lahore 54000, Pakistan

are tagged with I tag. All the other tokens are tagged with O which means that they are not part of any phrase chunk. The IOB tagging scheme works very similar to part of speech (POS) tagging. In POS tagging, each word is assigned with a single POS tag, but IOB-tagging marks phrases in a sentence. The chunking is usually performed after the part of speech tagging phase.

This paper presents the neural phrase chunking along with the development of an IOB-tagged corpus for Urdu. For the development of the corpus, CLE Urdu Digest POS Tagged corpus [11] has been used which contains 35 POS tags. In the annotation of the IOB tagged corpus, four phrases have been annotated, which are NP (noun phrase), VP (verb phrase), PP (post-positional phrase) and PRP (prepositional phrase). The annotation guidelines have been developed after linguistic analysis of the syntactic structure of the language which cover all the aspects of these phrases within the corpus. The tagging guidelines are also written to make the tagging task clear and more deterministic. A tagging utility has been developed which disseminates POS tagged text to assist the annotators. Annotation challenges were undertaken collectively, and guidelines were updated accordingly throughout the annotation process. The validation of the corpus is also a crucial task. The tagging task has been divided into batches, and the inter-annotator agreement has been computed for all batches. A minimum tagging similarity of 95% was ensured for each batch.

The neural chunker is based on long–short-term memory (LSTM) networks. LSTM networks are quite capable to learn sequence labels. However, neural models require a lot of annotated data to produce better results in terms of accuracy and f-scores. The developed dataset contains about one hundred thousands annotated words; therefore, we performed transfer learning by training word representations on a larger text corpus. For that purpose, we have trained context-free as well as contextualized word representations. The contextualized word representations outperformed the context-free word embeddings. The details of the chunking model and word representations are presented in Sect. 5.

The rest of the paper is distributed as follows. Section 2 describes the works related to phrase chunking. Section 3 provides the detailed annotation guidelines with the help of examples. Section 4 presents the annotation methodology, evaluation and statistics of the annotated corpus. Section 5 describes the chunking model and word representations. The chunking results are presented in Sect. 6. Finally, Sect. 7 concludes the paper by describing the findings.

2 Related Works

IOB tagged corpora have been developed for many languages through the years. This section describes the development of corpora and automatic phrase chunking. A noun phrase chunker for Urdu language has been developed in [12]. The corpus contained 101,414 words and 4,585 sentences. The corpus was manually annotated with noun phrases using IOB tagging scheme. The noun phrases follow several rules such as a noun and pronoun make minimal noun phrase, and all modifiers before a noun become part of the noun phrase. The data of 4,055 sentences including 91,429 words were used for training, and the remaining 530 sentences with 9,985 words were used for testing the chunker. A statistical technique based on hidden Markov model (HMM) was developed to generate automatic NP chunks. The results of all experiments showed an accuracy of 97.61% which is quite high as the dataset only contains NPs. They further implemented a hybrid approach for verb phrase chunking of Urdu which was also based on HMM along with a post chunking rule set in [13]. The Urdu verb phrase also has complex structures which include conjunct verbs, compound verbs and there are several auxiliary verbs within a phrase. Auxiliary verbs describe the aspect, modality, progress and tense of the verbs. They have further developed a manually annotated dataset containing one hundred thousand words. By using the hybrid method, they reported a tagging accuracy of 98.44%. A rule-based chunker has been proposed for Hindi in [14]. The handcrafted rules were derived for noun phrase, verb phrase, adverb phrase and conjunct phrase. They have developed a chunked corpus of five hundred sentences. The rule-based chunker performed with recall, precision and f-score of 69.0, 78.0 and 74.0, respectively.

A hybrid method has been introduced for chunking Turkish language in [15]. An annotated corpus containing eight thousand sentences was used for training and testing. The experiments were carried by using two chunk labels. Word2Vec [16] embeddings were trained on a large corpus containing five million sentences and were used for training support vector machines (SVM)-based chunker. The chunker performed with an f-score of 70.3 with vector size of 20.

A chunker for Korean language was developed by combining handcrafted rules and a machine learning method [17]. The handcrafted rules accelerated the performance of the chunker. Three basic chunks including noun phrase (NP), verb phrase (VP) and adverb phrase (ADVP) were described using IOB tagging scheme. The noun phrase incorporated nouns, pronouns, determiners and post-positions showing possessions and appearing in the middle of nouns. Moreover, a simple relative class without any sub-constituent also made a noun phrase. For verb phrases, 29 rules were used for chunk identification. The sequence of adverb phrase formed an adverb phrase. Multiple experiments were executed that also calculated the accuracy of handcrafted rules and machine learning methods separately. The accuracy of handcrafted rules was compared with machine learning method. The rule-based chunking gave an accuracy of 97.99% and an f-score of

91.87%, while machine learning method gave an f-score of 91.38%. The statistics evidently show that handcrafted rules accelerated the accuracy more than machine learning method for Korean.

A Vietnamese chunker has been presented in [18]. They developed an annotated corpus which contains nine thousand sentences. They have annotated noun phrases only. Three different techniques have been used to develop three chunkers which include, conditional random fields (CRF), support vector machines (SVM) and online passive-aggressive learning (PA). CRF-based model performed better as compared to other methods.

The parser, GTA (Granska Text Analyzer), for Swedish language has been developed in [19]. Multiple handcrafted rules were derived to mark the phrases of Swedish language including noun phrases (NP), verb chains (VC) and limited verb phrases, prepositional phrases (PP), adverb phrases (ADVP), adjective phrases (AP) and infinitive verb phrases (INFP). The noun phrase (NP) included minimal noun phrases (proper names and pronouns), complex noun phrases and coordinated noun phrases. The prepositional phrases were not included in the noun phrase. The verb phrase included simple and complex verbs. The prepositional phrases included prepositions that are followed by noun phrases. Moreover, the adverb phrases contained singleton adverbs, while adjective phrases included simple adjectives and a group of adjectives. All infinitive verb phrases start with infinitival marker that was followed by infinitive verb and optional noun phrase. The IOB tagging scheme has been used to mark the data. The rule-based chunker performed with a tagging accuracy of 88.4%.

Due to lack of language processing tools for Arabic text, a support vector machine-based model has been presented in [20]. This approach is based on automatically tokenization, POS tagging as well as base phrases (BPs) chunker. In Arabic language, conjunctions, prepositions and pronouns are cliticized. Moreover, base phrase chunking is the process of making different phrases including noun phrases, verb phrases, prepositional phrases and adjectival phrases. In tokenization, each word in the corpus was tagged with its morphological identity. The POS tag set used 24 tags which were taken from collapsed Penn Arabic Treebank. Furthermore, in base phrase chunking nine types of chunks were recognized using IOB tagging scheme. By using ChunkLink software, training data from Arabic treebank were derived. The treebank includes 4,519 sentences. The training, development and test set were same at all the steps. By using metrics approach, SVM tokenizer achieved 99.12% accuracy, while POS tagger and base phrase chunker achieved accuracies of 95.49% and 92.08%, respectively. These results are comparable to English text because both are trained on same sized corpus.

Like other languages, the automatic chunker has also been developed for ten South African languages [21]. In NCHLT Text Phase II project, 15,000 tokens were annotated with phrasal constituency at first step, and in second step, the automatic chunker for South African languages was developed. Five phrases that are, noun phrase, verb phrase, adjective phrase, adverb phrase and prepositional phrases were annotated which are based on CONLL-2000 shared task [22]. For data annotation, IOB tagging scheme was adapted which was assisted by Linguistic Annotation and Regulation Assistant (LARA3). As annotation by hand was difficult so by using LARA3 tool annotators generates accurate annotations. The click and highlight functionality helped annotators to assign the words to different phrases. In second step, three different automatic chunkers based on manual annotation for each language were developed. The chunkers were evaluated by using f-measures, and the results are quite promising.

Besides the described languages, the phrase chunking has been performed for many other languages including, Arabic (noun phrases) [23], Kannada [24], Chinese [25,26], Bengali [27], Marathi [28], Japanese [29], Thai [30], Manipuri [31] and Burmese [32]. The phrase chunking is an important task to perform shallow syntactic analysis. It replaces the constituency [33–35] and dependency parsing [36] when deeper analysis are not required. In this paper, we present the development of an IOB-tagged corpus and the a neural phrase chunker for Urdu by employing transfer learning.

3 Annotation Guidelines

The representation of chunks is done by trees or tags. The chunks are nonoverlapping phrases; therefore, IOB tagging scheme is widely used to annotate them. IOB stands for I(inside), O (outside) and B (beginning of the chunk) and each token in a sentence should belong to any one of these tags. Before marking IOB tags, POS tags are marked. In the POS tagging, each word is tagged with a single tag and the same case with IOB tagging. The POS tags are helpful to annotate the chunk labels as well as the learning process. A simple example of the annotated sentence is shown in (1). The annotation presents the transliteration of words, glasses, POS tagging, IOB labels and the translations. In the examples, Urdu words have been transliterated by using a transliteration scheme proposed in [37].

(1)

	vO	xuS	hE
	pron.3.Sg	happy	be.Pres.3.Sg
	PRP	NN	VBF
	B-NP	B-NP	B-VP

'He/she is happy'

In (1), there are three chunk phrases, two noun phrases and one verb phrase. The pronoun *vO* 'he/she' makes a simple noun phrase containing one word. Similarly, the word *xuS* 'happy' is also annotated as an NP. The verb *hE* 'be.Pres.3.Sg' has been annotated as a VP. In the annotation of the corpus, we have annotated four phrases which are noun phrase, post-positional phrase, prepositional phrase and verb phrase. A phrase is a unit which is a combination of words giving a sense [38]. Following sections describe the rules to annotate the Urdu phrases.

3.1 Noun Phrases (NPs)

A noun and pronoun can be a noun phrase or a group of words including nouns, pronouns and adjectives [12]. In other words, a noun phrase is the combination of words where the head word is a noun. The head word is the most prominent word which contains the sense of the whole chunk. Following sections present the examples of different types of noun phrases.

3.1.1 Common Nouns

A noun which is common to everyone and does not refer a particular person, thing or place, i.e., girl, boy, etc. The common nouns also include collective and abstract nouns. Some examples of Urdu common nouns are *pAnI* 'water,' *kitAb* 'book,' *yAd* 'memory,' etc. A minimal noun phrase may contain a single noun. In example (2), *kitAb* 'book' is a single word noun phrase.

(2)	kitAb	kA	nAm
	book.Nom.Sg.Fem	case.Gen.Sg	name.Nom.Sg.Masc
	NN	PSP	NN
	B-NP	B-PP	B-NP

'Title of the book'

A noun phrase is also formed by a noun to which different dependents are attached. The nature of a noun phrase is recursive because different dependents are attached to the head element. The most important dependents in noun phrases are modifiers. These are optional elements in noun phrases which means that without these elements, a noun can also provide complete meanings [39]. The nominal modifiers give information about the nouns. These include adjectives, cardinal, ordinal and quantifiers. Urdu modifiers are divided into adjectives (JJ), e.g., *acHA* 'good,' ordinal (OD), e.g., *dUsrA* 'second,' cardinal (CD), e.g., *dO* 'two,' quantifiers (Q), e.g., *kucH* 'some,' multiplicative (QM), e.g., *gunA* 'times' and fraction (FR), e.g., *AdHA* 'half.' When noun is preceded by all these elements then they make an NP [11]. Moreover, all adjectives which appear before nouns are the part of noun phrases. Examples (3) and (4) show the annotation of such noun phrases.

(3)	acHA	nAStA	karEN
	fine.Adj.Sg.Masc	breakfast.Nom.Sg.Masc	do.Pres.Pl
	JJ	NN	VBF
	B-NP	I-NP	B-VP
	gE		
	Fut.Pl		
	AUXT		
	I-VP		

'Will have a fine breakfast'

Example (3) presents a canonical noun phrase *acHA nAStA* 'fine breakfast' which has an adjective *acHA* 'fine' followed by a common noun *nAStA* 'breakfast.' The phrase has been annotated by using two tags which are B-NP and I-NP. B-NP represents that it is the start of an NP, and the second tag I-NP shows that the word is inside the running noun phrase. The end of a chunk occurs when the beginning of a next phrase appears, e.g., B-VP. Similarly, there is a verb phrase which also has two words *karEN gE* 'will do.' The annotation of verb phrases is presented in Sect. 3.2.

(4)	vahAN	kucH	lOg	tHE
	there	some	people	be.Perf.3.Pl
	NN	Q	NN	VBF
	B-NP	B-NP	I-NP	B-VP

'There were some people'

Example (4) shows the annotation of two noun phrases followed by a verb phrase. The word *vahAN* 'there' is a spatio-temporal noun which is also annotated as a common noun. Another noun phrase *kucH lOg* 'some people' make a noun phrase which contains a quantifier *kucH* 'kucH' followed by a common noun *lOg* 'people.'

3.1.2 Proper Nouns

A proper noun which is not common to every person, place, things, i.e., Lahore, Ali, Pakistan, etc. The proper nouns also make noun phrases as mentioned in example (5).

(5)	urdU	,	panjAbI	Or	fArsI
	Urdu	,	Punjabi	and	Persian
	NNP	PU	NNP	CC	NNP
	B-NP	I-NP	I-NP	I-NP	I-NP

'Urdu, Punjabi and Persian'

In example (5), there is only one phrase which has three proper nouns, a punctuation and a coordinate conjunction.

Interestingly, the punctuation symbol ',' and the conjunction *Or* 'and' have been marked as a part of the noun phrase. In the annotation, such combined phrases are marked as single noun phrases.

3.1.3 Compound Nouns

A noun phrase is also a combination of compound nouns or a combination of noun+verb construction. In Urdu, compound nouns make a noun phrase as mentioned in example 6.

(6)	laRkE	nE	intizAr	kiyA
	boy.Nom.Sg.Masc	case.Erg	wait	do.Perf.Pl.Masc
	NN	PSP	NN	VBF
	B-NP	B-PP	B-NP	B-VP

'The boy waited'

Example (6) shows the annotation of noun *intizAr* 'wait' which makes a complex predicate structure with the verb. However, these kinds of nouns give the verbal sense, but they do not have the forms like verbs. Therefore, they have been annotated like common nouns making single word noun phrases mostly. The example also shows a post-positional phrase (PP) which annotates a ergative case marker. The annotation of post-positional phrases is discussed in Sect. 3.3.

3.1.4 Pronouns

The pronouns appear as a replacement of nouns. The Urdu pronouns are categorized in seven categories including personal, demonstrative, possessive, relative demonstrative, relative personal, reflexive and reflexive APNA pronouns [11]. Following sections present the annotation of different types of pronouns with the help of some examples.

a) *Personal, Reflexive and Relative Personal Pronouns.*
A noun phrase can have a single pronoun in it. The single pronouns can be personal, reflexive and relative personal pronouns [12].

(7)	vO	sakUl	gayA
	pron.3.Sg	school.Nom.Sg	go.Perf.Sg.Masc
	PRP	NN	VBF
	B-NP	B-NP	B-VP

'He went to school'

Example (7) shows the annotation of a personal pronoun *vO* 'he' which has been annotated as a single word noun phrase. Similarly, example (8) presents a relative personal pronoun *jO* 'which' and its annotation. It relates a preceding noun *kAm* 'work.' Such pronouns are also marked as single word noun phrases.

(8)	vO	kAm	jO	kal	hO
	pron.3.Sg	work.Sg.Masc	which	yesterday	be.Sg
	PDM	NN	PRD	NN	VBF
	B-NP	I-NP	B-NP	B-NP	B-VP
	gyA				
	go.Perf.Sg.Masc				
	AUXA				
	I-VP				

'The work which was accompished yesterday'

Noun phrases sometimes can also consist of two pronouns. Both pronouns are marked as an NP on the basis of contextual information. Moreover, two pronouns when preceded by a noun also make a single noun phrase as mentioned in example (9).

(9)	mujHE	xud	jAnA	paRE	gA
	pron.1.Sg	myself	go.Inf.Sg.Masc	keep.Pres.Pl	Fut.Sg
	PRP	PRF	VBI	AUXA	AUXT
	B-NP	I-NP	B-VP	I-VP	I-VP

'I myself will go'

Example (9) shows the annotation of a noun phrase containing two types of pronouns, personal and reflexive *mujHE xud* 'I myself.'

b) *Demonstrative, Relative Demonstrative, Possessive and Reflexive APNA Pronouns.*
Pronouns also act as determiners when appearing before nouns. There are various types of pronouns in Urdu as mentioned above, but demonstrative, relative demonstrative, possessive and reflexive APNA pronouns make a noun phrase when preceded by a noun [12].

(10)	hAmid	yE	haqIqat
	Hamid.Nom.Sg.Masc	pron.2.Sg	fact.Nom.Sg.Fem
	NNP	PDM	NN
	B-NP	B-NP	I-NP
	jAn	gyA	
	know	go.Perf.Sg.Masc	
	VBF	AUXA	
	B-VP	I-VP	

'Hamid came to know this fact'

In example (10), the noun phrase *yE haqIqat* 'this fact' contains a demonstrative personal pronoun *yE* which is a determiner for the noun *haqIqat* 'fact.' Similarly, the noun phrases like *jO lOg* 'which people,' *mErA bastA* 'my bag' and *apnA kAm* 'my/your work' have relative demonstrative, possessive and reflexive APNA pronouns, respectively.

3.1.5 Noun Complex Predicates

Urdu language employs a complex predicate structure. The structure of complex predicates is complex as it includes V+V (verb + verb), Adj+V (adjective + verb) and N+V (noun + verb) constructions [4,5]. The nouns in complex predicates are also marked as single word noun phrases as shown in (11).

(11)

aslam	nE	kAm
Aslam.Nom.Sg.Masc	case.Erg	work.Nom.Sg.Masc
NNP	PSP	NN
B-NP	B-PP	B-NP

xatam	kiyA
finish	do.Perf.Sg.Masc
NN	VBF
B-NP	B-VP

'Aslam completed the work'

Example (11) demonstrates the annotation of a complex predicate structure by using a noun *xatam* 'finish' which is followed by a light verb *kiyA* 'do.Perf.Sg.Masc.' Complex predicates usually include the light verbs as the meaning is represented by a noun, adjective or quantifier. In the annotation process, such nouns have been annotated as single word noun phrases.

3.1.6 Noun Phrases with *vAlA*

vAlA is a suffix and conjoins with nouns and verbs. The VALA tag when occurs with noun and verb, it becomes the part of a noun phrase as shown in example (12) [40].

(12)

dudH	vAlA
milk.Nom.Sg.Masc	vAlA
NN	VALA
B-NP	I-NP

'The milkman'

3.1.7 Punctuation

A comma is considered as conjunction when occurs between nouns. Moreover, quotation marks also become part of noun phrases when make sense [41]. An example of double quotes is shown in (13).

(13)

mArUf	kitAb	"	TAem
famous	book.Nom.Sg.Fem	punc	time.Nom.Sg.Masc
JJ	NN	PU	NNP
B-NP	I-NP	I-NP	I-NP

"
punc
PU
I-NP

'The famous book "Time"'

Example (13) shows a noun phrase which has a proper noun *tAem* 'Time' surrounded by double quotes. These types of quotes have been annotated as part of noun phrases.

3.1.8 Adpositions

Adposition is a combination of prepositions and post-positions. Both grammatical categories play important role in noun phrase annotation [6]. Post-positions, which represent the range, will be marked as a part of an NP. Example (14) shows a post-position *tA* 'till' representing a time duration from one month to another. These kinds of post-positions usually appear between two nouns hence marked as part of noun phrases.

(14)

13	aiprel	tA	25
thirteen	April.Nom.Sg.Masc	till	twenty-five
CD	NNP	PSP	CD
B-NP	I-NP	I-NP	I-NP

aiprel
April.Nom.Sg.Masc
NNP
I-NP

'From 13 to 25 April'

3.2 Verb Phrases (VPs)

Verb is a word which shows action, state or event in a sentence. Urdu verbs are differentiated into finite, infinitive, light, copula and auxiliary verbs [11]. Urdu verbs also take different causative and double causative forms [3]. Finite verbs define the meanings of sentences and are marked with VBF POS tag, whereas infinitive verbs are differentiated by using VBI tag. The light verbs are annotated as finite verbs. Auxiliary verbs are used to represent aspect, modality, progress and tense. There are four types of auxiliaries including tense (AUXT), modal (AUXM), aspectual (AUXA) and progressive auxiliary (AUXP). Auxiliary verbs usually add nuance to the main verb [11]. Copula verbs have been annotated as tense auxiliaries. A verb phrase is a group of words

which does not contain subject and predicate as well as acts as a verb [42]. Following sections present the annotation of verb phrases with the help of some examples.

3.2.1 Verb Complex

Verb complex includes main verb as well as auxiliary verbs. A simple verb phrase consists of a single main verb as mentioned in example (15) [13]. The main verb *AE* 'come.Perf.Pl.Masc' shows the action of the sentence hence annotated as a single word verb phrase by using the B-VP label.

(15)	vO	lAhOr	AE
	pron.3.Pl	Lahore.Nom.Sg.Masc	come.Perf.Pl.Masc
	PRP	NNP	VBF
	B-NP	B-NP	B-VP

'They came to Lahore'

Urdu auxiliary verbs appear after main verbs in contrast with English where auxiliary verbs occur before the main verb [42]. An example of such verb phrase is shown in (16).

(16)	us	nE	kAm		xatam
	pron.3.Pl	case.Erg	work.Nom.Sg.Masc		finish
	PRP	PSP	NN		NN
	B-NP	B-PP	B-NP		B-NP
	kar	liyA		hE	
	do.Pres.Sg	take.Perf.Sg.Masc		be.Pres.Sg	
	VBF	AUXA		AUXT	
	B-VP	I-VP		I-VP	

'He has completed the work'

A verb phrase also consists of main verb which is followed by a tense auxiliary as mentioned in example (17). It shows a main verb *jAtA* 'go.Pres.3.Sg.Masc' followed by an auxiliary verb *hE* 'be.Pres.Sg.' Both have been annotated in a verb phrase.

(17)	vO	sakUl	jAtA
	pron.3.Sg	school.Nom.Sg.Masc	go.Pres.3.Sg.Masc
	PRP	NN	VBF
	B-NP	B-NP	B-VP
	hE		
	be.Pres.Sg		
	AUXT		
	I-NP		

'He goes to school'

3.2.2 Infinitive Verbs

Infinitive verbs act as verbal noun and display a syntactic distribution that is different from the main verb [11]. The infinitive is the form of verb which takes suffix *nA* that may also be inflected to have suffixes *nI* and *nE*. Some examples of infinitive verbs are, *karnA* 'to do,' *kHAnA* 'to eat' and *paRHnA* 'to read,' etc. The infinitive verbs show obligation, permission, negative assertion and purpose. An infinitive verb also makes a verb phrase. Example (18) shows annotation of an infinitive verb.

(18)	kAm	karnE	kE
	work.Nom.Sg.Masc	do.Inf.Pl	case.Gen
	NN	VBI	PSP
	B-NP	B-VP	B-PP
	liyE		
	take.Perf.Pl.Masc		
	PSP		
	I-PP		

'To do the work'

In (18), the verb *karnE* 'do.Inf.Pl' is marked with a POS tag VBI which is further annotated as a verb phrase. The phrase *kE liyE* 'for' has been annotated as a compound post-positional phrase as it contains two post-positions.

3.2.3 Copula Verbs

The tense auxiliaries become copula verbs when function as main verb. The tense auxiliaries are merged with copula verbs because both verbs share same surface forms. The tense auxiliaries always come with main verb and provide tense information. An example of copula verbs is given in (19).

(19)	yE	Ek	azIm	dAstAn	hE
	pron.2.Sg	one	great	story.Nom.Sg.Fem	be.Pres.Sg
	PRP	CD	JJ	NN	VBF
	B-NP	B-NP	I-NP	I-NP	B-VP

'This is a great story'

In example (19), the verb *hE* 'be.Pres.Sg' is a copula verb and has been annotated as a verb phrase. If the word *hE* appears as auxiliary following the main verb then it is marked with a POS tag AUXT and as part of the verb phrase but not the copula verb.

3.2.4 Complex Predicates

A complex predicate is a term that involves two or more elements (e.g., verbs, nouns and adjectives) to become a

predicate structure. All these elements except nouns when combine with main verbs make a verb phrase in the annotation. If a complex predicate has an adjective or quantifier, they are part of the verb phrase. For example, the adjective *priSAn* 'worried' in (20) makes a verbal sense when combined with the light verb *hOnA* 'be.Inf.Sg.Masc' hence annotated as a verb phrase.

(20)
priSAn	hOnA
worried	be.Inf.Sg.Masc
JJ	VBI
B-VP	I-VP

'to get worried'

(21)
kam	kar	diyA
less	do.Pres.Sg	give.Perf.Sg.Masc
Q	VBF	AUXA
B-VP	I-VP	I-VP

'has been reduced'

Similarly, in example (21), the quantifier *kam* 'less' has been annotated as part of the verb phrase as it makes a complex predicate structure with the light verb *kar* 'do.Pres.Sg.'

3.2.5 Negation

The negation particles usually occur with the verbs so are grouped in verb phrases [41]. Example (22) shows a verb phrase which has a negation+verb structure. The negation particle *nahIN* 'no/not' has been annotated as part of the verb phrase. The negation particle will be marked as out of a phrase tag (O) if it does not appear with the main verb. However, in the grammatical structure, the negations usually appear at pre-verbal positions.

(22)
is	kI	kOI	misAl
pron.3.Sg	case.Gen.Fem	any	examle.Nom.Sg.Fem
PRP	PSP	PDM	NN
B-NP	B-PP	B-NP	I-NP

nazar	nahIN	AtI
see	negation	come.Pres.3.Sg.Fem
NN	NEG	VBF
B-NP	B-VP	I-VP

'There is not any example of it'

3.3 Post-Positional Phrases (PPs)

Post-positions follow nouns or pronouns and mark the case information. Urdu post-positions function similar to prepositions of European languages. Post-positions are also referred as case markers. There are eight cases [6] which are represented by post-positional clictics as shown in Table 1.

Table 1 Urdu case markers

Sr#	Case	Case Marker
1	Nominative	Ø
2	Ergative	nE
3	Accusative	kO
4	Dative	kO
5	Instrumental	sE
6	Ablative	sE
7	Locative	meN, par
8	Genitive	kA, kI, kE

3.3.1 Single Post-position

Post-positional phrases consist of single post-positions, i.e., *kA, kI, kE* and with combination of other words such as *kI jAnib sE* 'from someone' *kI binA par* 'due to.' Both of these scenarios make post-positional phrases. Example (23) shows a post-positional phrase (PP) which annotates the genitive case marker *kA* 'case.Gen.Sg.'

(23)
mAliyAt	kA	mehkmA
finance.Nom.Pl	case.Gen.Sg	department.Nom.Sg.Masc
NN	PSP	NN
B-NP	B-PP	B-NP

'The department of Finance'

3.3.2 Compound Post-positions

Post-positional phrases consist of a single as well as multiple words. Compound post-positions consist of inflected *kA* + a noun, an adjective, an adverb, i.e., *kE zariE* 'by' or other complex construction such as *kI vajA sE* 'due to which.'

(24)
mOjzA=e	xudAvandI
miracle.Nom.Sg.Masc	God.Nom.Sg.Masc
NN	NNP
B-NP	I-NP

kE	tOr	par
case.Gen.Pl	case	case.Loc
PSP	PSP	PSP
B-PP	I-PP	I-PP

'As God's miracle'

Example (24) presents the annotation of a compound post-positional phrase which is *kE tOr par* 'like/as.' The POS tag set marks all types of case markers by using PSP tag hence annotated as a single PP phrase.

3.4 Prepositional Phrases (PRPs)

Prepositions make prepositional phrases in the annotation. Prepositions are very rare in the corpus. Some examples of Urdu prepositions are; *fI* 'in/per,' *az* 'from' , *sivAE* 'except' and *bajuz* 'except,' etc. Prepositions have been annotated as prepositional phrases (PRP). Example (25) shows annotation of a prepositional phrase.

(25)	sivAE	alI	kE
	except	Ali.Nom.Sg.Masc	case.Gen.Pl
	PRE	NNP	PSP
	B-PRP	B-NP	B-PP

'Except Ali'

In (25), the word *sivAE* 'except' appears before the proper noun *alI* 'Ali' which is marked by a POS tag PRE. All such prepositions are marked as prepositional phrases in the annotation process.

3.5 Outside the Phrase

In the IOB tagging, each token should belong to a phrase or chunk. But some tokens or words does not fall in any category. The annotation of the IOB tagged Urdu corpus contains four phrases. Therefore, the words which do not belong to any specific phrase are marked with label 'O' (Outside the phrase tag). The constructions annotated with 'O' label include, adverbs, subordinate and coordinate conjunctions, punctuation and list indices, etc. Example (26) shows an example where an adverb *zarUr* 'definitely' has been marked as outside of the phrase.

(26)	vO	pIcHE	zarUr	reh
	pron.3.Pl.Masc	behind	definitely	live.Pres.Sg
	PRP	NN	RB	VBF
	B-NP	B-NP	O	B-VP
	gaE			
	go.Perf.Pl.Masc			
	AUXA			
	I-VP			

'They definitely left behind'

Cardinals are numbers which are sometimes used for indexing and serial numbers, etc. In English language, serial numbers or cardinals are marked with LST tags. On the other hand, in Urdu language, index numbers are marked as outside the phrase. An example is shown by (27) where the number '1' followed by a colon has been marked with 'O' label.

(27)	1	:	apnE	kAm	par	tvajA
	one	collon	yours	work.Nom.Sg	case.Loc	focus
	CD	PU	APNA	NN	PSP	NN
	O	O	B-NP	I-NP	B-PP	B-NP
	dEN					
	give.Pres.Pl					
	VBF					
	B-VP					

'1: Focus on your work'

Urdu language also has a property to have duplication of words. This duplication is used to put stress in one's point. Such words which come in duplicate form in the corpus are marked outside the phrase. An example is given by (28) where the words *alag alag* provide the meaning of an adverb hence annotated as outside the phrase.

(28)	jildEN	alag	alag
	volume.Nom.Pl.Fem	separate	separate
	NN	JJ	JJ
	B-NP	O	O
	tHIN		
	be.Perf.Pl.Fem		
	VBF		
	B-VP		

'Volumes were separated'

Subordinating conjunctions are used to join two clauses, one is dependent clause, and other is independent clause. In Urdu language, the function of subordinating clause is the same as they are used in English language. Similarly, coordinating clitics are used as conjunctions between clauses as well as nouns, adjectives, etc. When coordinate conjunctions appear between clauses, they are annotated as outside the phrase. Example (29) presents the annotation of a subordinate conjunction *tAkE* 'so that' when has been marked as outside the phrase.

(29)	tarjumA		diyA	
	translation.Nom.Sg.Masc		give.Perf.Sg.Masc	
	NN		VBF	
	B-NP		B-VP	
	hE	tAkE	AsAnI	rahE
	be.Pres.Sg	so	ease.Nom.Sg.Fem	live.Pres.Sg
	AUXT	SC	NN	VBF
	I-VP	O	B-NP	B-VP

'The translation is provided so that it helps'

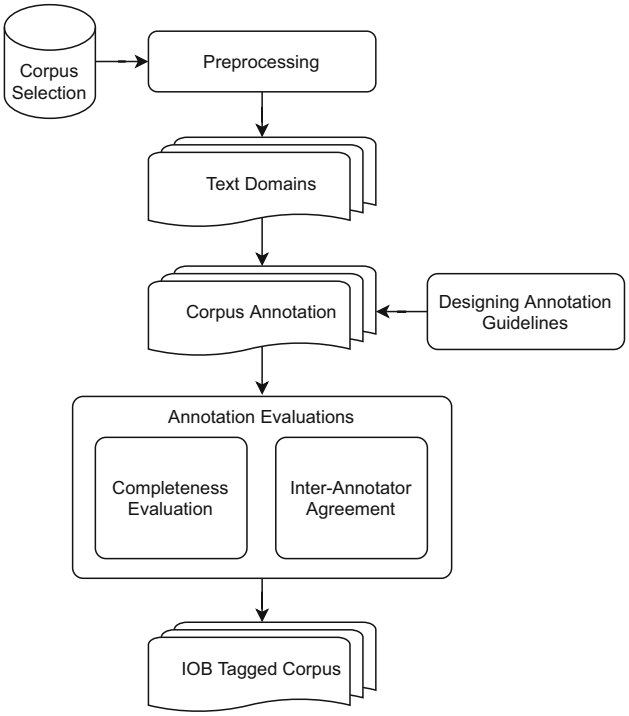

Fig. 1 IOB tagged corpus annotation methodology

There are number of punctuation marks present in different languages. Punctuation marks consist of commas, brackets, full stop, etc. In the annotation, all such punctuation marks are considered as outside of any phrase. In short, if a token does not appear to be part of four selected phrases, it is marked with outside label.

4 Corpus Annotation

In this section, we discuss the annotation methodology. The whole annotation process has been completed semi-automatically. Figure 1 shows the block diagram of annotation process for the development of IOB tagged corpus.

The first step is the corpus selection for the annotation. Preprocessing is an important step after corpus selection which ensures the cleaned data and work plan for the annotation. The corpus text belongs to fifteen different domains. A comprehensive annotation guidelines have been developed before the actual annotation process. To ensure the correctness, completeness and consistency of the corpus, a multi-step evaluation process has been carried out throughout the annotation. Next sections discuss these steps in more detail.

4.1 Corpus

The corpus is designed in such a way that it covers a number of text domains. To develop IOB tagged corpus for Urdu, CLE Urdu Digest POS Tagged Corpus 100K [11] has been used which contains one hundred thousands words. The corpus contains the text from 15 different text domains like entertainment, sports, culture, health, letters, novels, etc. Table 2 shows the details of the corpus with respect to number of sentences and tokens. After selecting the corpus, two tasks have been performed which include cleaning and word segmentation process. The corpus was already tagged with parts of speech tags. For this purpose, CLE POS tagset is used which is comprised of 35 tags. After getting POS tagged corpus, the phrases were determined for IOB tagging. The annotated corpus is available online[1]. Detailed IOB annotation guidelines are presented in Sect. 3.

4.2 Preprocessing

Two tasks have been performed as the preprocessing step, annotation utility employment and time estimation. A tagging utility has been employed to annotate the corpus. Rather than typing the tags using text editing software, the utility facilitates the annotators to add the tags just by clicking the buttons for that specific tag. Each button corresponds to an IOB tag. The utility helped in annotation process because it increased the annotation efficiency and minimized the rate of errors. Annotators put the cursor on desired place and then click on the relevant tag and the tag automatically marks the word. For example, if an annotator wants to mark B-NP, just click the B-NP button and the tag will be automatically marked on the desired position.

The second task was the time estimation for the annotation. An activity was performed by the annotators to estimate the annotation time with respect to the number of tokens. During this activity, annotators have annotated the text for one hour. On the basis of this activity, batch sizes have been determined. It was calculated that each annotator can annotate 600 words per hour, which means each annotator can annotate 4,200 words per day.

4.3 Manual Annotation

Two linguists have annotated the corpus by using the IOB tag set. Each annotator gets the batch which they had to return back after marking IOB tags. During the annotation process, annotators face various complex chunk structures, determination of complex predicates, brackets, POS, etc. Collected decisions have been made to resolve such types of ambiguities. Evaluation process has been carried out throughout the annotation.

[1] https://cle.org.pk/clestore/urduphrasechunker.htm

Table 2 Domainwise statistics of the corpus

Sr#	Domains	#Sentences	#Tokens
1	Book Reviews	194	4,445
2	Culture	421	8,131
3	Education	267	5,828
4	Entertainment	248	5,047
5	Health	534	9,594
6	Interviews	577	11,680
7	Letters	650	11,267
8	Novels	225	4,370
9	Press	424	9,779
10	Religion	488	9,354
11	Science	405	8,294
12	Short Stories	445	6,768
13	Sports	505	9,933
14	Technology	124	2,464
15	Translation of Foreign Languages	281	5,079
	Total	5,788	112,033

Table 3 Sample error log sheet for completeness evaluation

Sr#	File Name	Sentence No.	Error
1	input-1.txt	86	;/PU
2	input-1.txt	86	aOr/CC
3	input-1.txt	89	;/PU
4	input-1.txt	89	aOr/CC
5	input-2.txt	79	;/PU
6	input-2.txt	158	/PU/I-NP
7	input-2.txt	172	aOr/CC
8	input-3.txt	249	//PU/O
9	...	...	...

4.4 Annotation Evaluation

After getting tagged data from the annotators, the evaluation process has been done in two steps. Firstly, completeness and correctness check has been applied and then consistency has been evaluated by computing the inter-annotator agreement for the reference corpus.

4.4.1 Completeness and Correctness

Completeness and correctness were checked through a computer utility automatically in which missing and incorrect IOB tags were identified. The utility automatically generated an error log sheet in which all the missing or incorrect IOB tags were mentioned. Then on the basis of this error log sheet, the text was sent back to the relevant annotator to correct the tags. A sample error sheet is shown in Table 3.

4.4.2 Inter-Annotator Agreement

Inter-annotator agreement has been calculated for each batch. For this purpose, 10% reference corpus has been randomly generated which was annotated by an additional linguist. The completeness and correctness have been checked for reference corpus as well. Figure 2 shows the flowchart of the

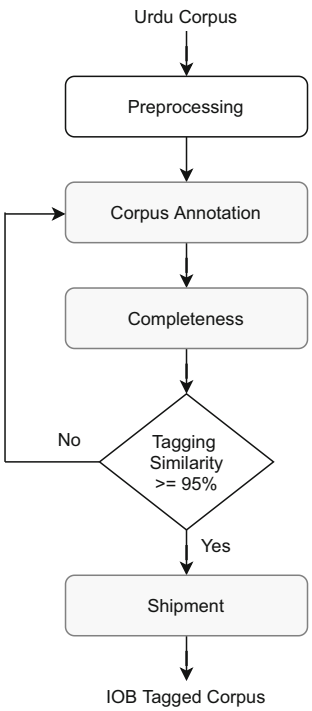

Fig. 2 Annotation evaluation process

Table 4 Inter-annotator agreement for three batch with respect to tagging similarity and f-scores

Category	Batch-1	Batch-2	Batch-3	Overall
Label Agreement (%)	95.3%	96.0%	96.1%	95.8%
Phrase Agreement (f-score)	94.48	94.59	94.54	94.53
Kappa Coefficient	0.9412	0.9499	0.9508	0.9478
No. of Sentences	137	171	212	520
No. of Tokens	3,240	4,046	4,090	11,376

evaluation process including completeness evaluation and inter-annotator agreement.

A batch is shipped if there is more than 95% tagging similarity of the reference corpus annotated by the annotator and the expert linguist. If the dissimilarities exceeds 5%, then the batch is sent back to the relevant annotator for reviewing. Errors and dissimilarities are remained hidden from the annotator to ensure unbiased revision. The corpus has been annotated in three batches, and the inter-annotator agreement has been calculated for them. Table 4 shows the inter-annotator agreement and number of sentences and tokens in the reference corpus.

Inter-annotator shows the tagging and syntactic understating of the annotators with respect to reference annotation. The agreement has been calculated by computing the label agreement, and phrase agreement and Kappa coefficient. For the computation of phrase (chunk) agreements, the f-scores have been computed with respect to phrases as show by Equation 5. For inter-rater agreement, Cohen's Kappa coefficient [43] has been computed for each batch. The coefficients have been calculated based on IOB tags. The interpretation of the coefficient values is important to derive any conclusion. The coefficient value of zero or less indicates poor agreement, 0.01 to 0.20 shows slight agreement, 0.21 to 0.40 means fair agreement, 0.41 to 0.60 represents moderate agreement. 0.61 to 0.80 shows substantial agreement, and the value between 0.81 and 1.00 indicates perfect agreement. The label agreement has been calculated by dividing the matched tags with all number of tags.

The label annotation agreement is 95.8% which is quite acceptable. The overall phrase agreement between annotators is an f-score of 94.53 which is quite acceptable when annotating a phrase structure of a language. The kappa coefficients are also quite high with an overall agreement of 0.9478 which shows perfect agreement with respect to IOB labels. These agreement scores indicate a collective understanding of the annotators to annotate the selective phrase chunks of Urdu.

4.5 Tagged Corpus Statistics

This section presents the detailed statistics of the annotated corpus with respect to annotated phrases and their frequencies. The corpus has been annotated for four phrases which include, noun phrases, post-positional phrases, verb phrases and prepositional phrases. All other constructs and tokens have been annotated as outside of the phrase by using the tag 'O.' Table 5 presents these phrases and their frequencies in the corpus.

Table 5 Corpus statistics with respect to phrases

Sr#	Phrase	Frequency	Coverage
1	NP	34,107	42.86%
2	PP	16,444	20.66%
3	VP	12,604	15.84%
4	PRP	8	0.01%
5	O	16,413	20.63%
	Total	79,576	–

Table 5 further shows the coverage of each phrase by presenting percentages. Noun phrases have highest frequency in the corpus which covers 42.86% of the annotations. Post-positional and verb phrases have the coverage of 20.66% and 15.84%, respectively. There are only eight occurrences of the prepositional phrases due to the nature of the language. It is important to note that the outside of phrase constructions also have sufficient number of annotations in the corpus. However, about 80% of the text has been annotated in any of the selected phrases. Table 6 shows the phrasal frequencies with respect to text domains.

Table 6 shows the counts of phrases within each domain. However, the number of sentences and tokens are different among domains. Therefore, we have analyzed the corpus by showing the percentages so that they are comparable with other domains. Table 7 shows the domainwise percentages against each phrase label.

All the domains have sufficient representation of each phrase except prepositional phrases. This analysis provides an insight of the corpus and its annotation. Some phrase annotations are quite related with each other. For example, higher frequency of noun phrases leads to higher frequent of post-positional phrases as shown against the domains of book reviews, press, culture, education and entertainment. Similarly, high frequency of noun phrases also leads to the low frequency of out of phrase labels and vice versa. The domains of book reviews and press have high number of noun phrases

Table 6 Domainwise statistics with respect to phrase counts

Sr#	Domain	NP	PP	VP	PRP	O
1	Book Reviews	1,412	731	463	3	553
2	Culture	2,669	1,339	949	2	1,186
3	Education	1,815	901	551	0	841
4	Entertainment	1,480	716	502	1	661
5	Health	2,846	1,324	1,100	0	1,444
6	Interviews	3,695	1,757	1,266	0	1,615
7	Letters	3,292	1,472	1,387	0	1,809
8	Novels	1,241	531	566	1	784
9	Press	2,937	1,599	957	0	1,148
10	Religion	2,993	1,406	1,189	1	1,440
11	Science	2,424	1,221	830	0	1,165
12	Short Stories	2,016	841	946	0	1,279
13	Sports	3,018	1,573	1,009	0	1,333
14	Technology	720	343	250	0	363
15	Translation of Foreign Languages	1,549	690	639	0	792
	Total	34,107	16,444	12,604	8	16,413

Table 7 Domainwise statistics with respect to phrase percentage

Sr#	Domain	NP (%)	PP (%)	VP (%)	PRP (%)	O (%)
1	Book Reviews	44.66	23.12	14.64	0.09	17.49
2	Culture	43.43	21.79	15.44	0.03	19.30
3	Education	44.18	21.93	13.41	0.0	20.47
4	Entertainment	44.05	21.31	14.94	0.03	19.67
5	Health	42.39	19.72	16.38	0.0	21.51
6	Interviews	44.34	21.08	15.19	0.0	19.38
7	Letters	41.36	18.49	17.42	0.0	22.73
8	Novels	39.74	17.0	18.12	0.03	25.10
9	Press	44.23	24.08	14.41	0.0	17.29
10	Religion	42.58	20.0	16.92	0.01	20.49
11	Science	42.98	21.65	14.72	0.0	20.66
12	Short Stories	39.67	16.55	18.61	0.0	25.17
13	Sports	43.53	22.69	14.55	0.0	19.23
14	Technology	42.96	20.47	14.92	0.0	21.66
15	Translation of Foreign Languages	42.21	18.80	17.41	0.0	21.58

and less number of outside of the phrase labels. The domains of novels and short stories have less number of noun phrases and high number of outside of the phrase labels. However, it can be concluded that the corpus has quite even coverage of each phrase across text domains. Table 8 shows the division of the annotated corpus into train, test and development sets.

The corpus division has been performed by using a standard 80:20 ratio. The 20% evaluation corpus was further divided into the test and development sets, each containing 10% of the text. Table 8 also presents the frequency of annotated phrases against train and evaluation sets. The training set contains 4,536 sentences with 88,426 tokens, and the test and development sets contain 526 sentences each. These sets have been used for the training and testing of the neural chunker which is presented in Sect. 5.

5 Neural Phrase Chunker

Recurrent neural networks (RNNs) provide a framework to learn the sequence labels by using contextual information of the whole sequence. Our chunking model has been developed on recurrent neural networks by using long–short-term memory (LSTM) networks. The LSTM layers have been used in bidirectional manner which are referred as BiLSTM networks. BiLSTM-based models are quite capable to learn and

Table 8 Division of the corpus into train, test and development sets

Category	Train	Test	Dev	Total
No. of Sentences	4,536	626	626	5,788
No. of Tokens	88,426	11,745	11,862	112,033
Noun Phrases	26,937	3,618	3,552	34,107
Post-Positional Phrases	13,005	1,725	1,714	16,444
Verb Phrases	9,896	1,347	1,361	12,604
Prepositional Phrases	8	0	0	8
Outside of Phrase	12,910	1,736	1,767	16,413

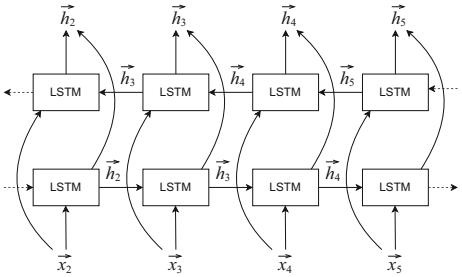

Fig. 3 Bi-directional long–short-term memory model for sequence labeling

predict sequence labels like IOB tags. Figure 3 shows the architecture of BiLSTM model by using two LSTM layers.

The conventional RNNs face the problem of vanishing gradient when trained on longer sequences which makes them less practical. LSTM layers solve this problem by providing an ability to forget the irrelevant information while keeping the required context in the memory of the network. An LSTM cell has hidden neural network layers which work as forgetting and remembering gates. The LSTM-based model provides better learning ability for a sequence labeling, and IOB tagging is a typical sequence labeling problem for a language. We have trained BiLSTM-based deep neural model to perform phrase chunking of Urdu. For a vector $w_1 : n$ at i, the $BiLSTM(w_1 : n, i)$ is represented by equation 1. Equation 1 show a vector i which uses the previous contextual information from w_1 to i and the upcoming context from w_i to n. After the LSTM layer, a dense layer was employed which used $softmax$ as nonlinearity function to perform multi-class classification. The $softmax$ layer finally predicted the IOB labels against each input sequence as represented by equation 2.

$$BiLSTM\,(w_{1:n}, i) = LSTM_f\,(w_{1:i}) \circ LSTM_r\,(w_{n:i}) \quad (1)$$

$$o_i = Softmax\,(Wh_i + b) \quad (2)$$

For a sentence of length n ($e.g.$, $x_1, x_2, ..., x_n$), each token is represented as an embedding vector. Similarly, the POS tag sequence ($t_1, t_2, ..., t_n,$) are learned against each token in the form of vectors. To include the information of POS tags along with language tokens, both word embedding vectors and POS embedding vectors were concatenated to produce a single training vector. For a token i, the word embedding vector $emb(w_i)$ and POS embedding vector $emb(t_1)$ produce a combined vector, i.e., $emb(w_i) \circ emb(t_i)$ after concatenation.

Figure 4 shows the overall model which has been trained for IOB tagging. First step shows the vector encoding of words and POS tags in the prepared dataset. The vector encodings are further given to input layer which concatenates the word embeddings, pre-trained word representations and POS representations to feed them to hidden layers of BiLSTMs. We have experimented up to three hidden layers. The contextual representations from hidden layers are further fed to the dense layer with $softmax$ classifier which produces the probabilities against each label. The labels with highest probabilities are considered as the predicted output against each instance of the evaluation set. We further performed the transfer learning to overcome the issue of data sparsity.

5.1 Transfer Learning

Neural models are capable to produce state of the art results, but they require a lot of training data for learning. It is quite costly to annotate a huge data set for any language. Therefore, transfer learning is a suitable approach to overcome data sparsity by training word representations on a large unannotated corpus. It also caters the out of vocabulary problem. For training a phrase chunker for Urdu, we trained word representations on a plain text corpus containing 35 million tokens [44]. We have trained context-free word embeddings as well as contextualized word representations. Sections 5.1.1 and 5.1.2 describe the mechanisms of these two types of word representations.

5.1.1 Context-Free Word Representations

Context-free word representations produce single feature vector for each token in the vocabulary. In short, each word has a single meaning associated with it. These representations are capable of learning syntactic and semantic information of a language. The well-known representations are GloVe [45] and Word2Vec [16] embeddings. Both representations produce feature vectors for each word but are implemented by using different algorithms. The GloVe embeddings are trained based on word to word co-occurrences in the whole corpus, whereas the Word2Vec uses co-occurrences of neighboring words. Word2Vec uses a local window to learn the context of a words. The Word2Vec representations can be trained by implementing two models either skip-gram or the continuous bag of words (CBoW) model. The skip-gram is a feed-forward neural network model which takes a word at input layer and predicts its context words within the selected window. On the other hand, the CBoW model takes the con-

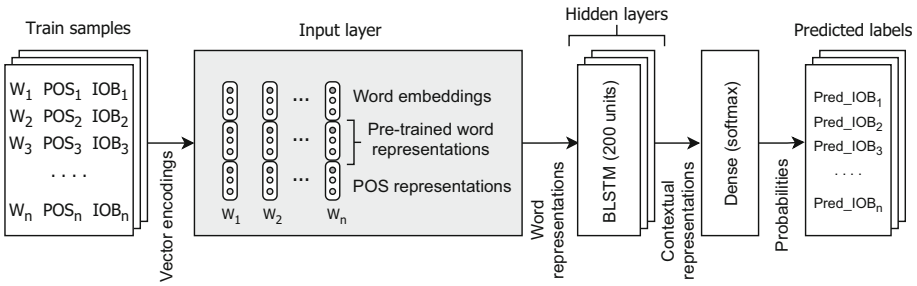

Fig. 4 BiLSTM-based model for IOB tagging

text words at input layer and predicts the original word. In both models, the feature vectors are achieved by giving the input and attaining the hidden layer values. In this paper, we have trained Word2Vec embeddings to perform transfer learning in the BiLSTM network-based phrase chunker. The vocabulary of the word embeddings contains 72 thousands words with 100 dimensions. The training has been done by selecting a window size of five words. Section 6 presents the chunking results by using Word2Vec word representations.

5.1.2 Contextualized Word Representations

The Word2Vec representations are context-free, i.e., Word2 Vec computes a single feature vector for a word. It does not analyze the meanings based on the use of the words. In natural languages, words bear different meanings based on their contexts. For the sequence labeling tasks like POS tagging and chunking, the contextual representations can be quite useful. Therefore, we have trained deep contextualized word representations (ELMo) as described in [46]. ELMo (Embeddings from Language Models) representations learn the word vectors from deep bidirectional language model. It combines the deep layers to produce the word representations. ELMo representations are character-based which are used to achieve the word vectors for out of vocabulary words and are helpful to capture the morphological insight. The pre-trained ELMo weights are used to compute the feature vectors for all words in a sentence. These vectors are computed on the bases of their context words in a sentence. The contextualization is helpful to perform word sense disambiguation and further improves the sequence labeling results. We have used a plain Urdu corpus containing 35 million tokens to train ELMo embeddings. To analyze the contextual representations, we have selected a tiny corpus as presented from (30) to (33).

(30) sAmp cUhE=kI
snake.Nom.Sg.Masc mouse.Sg.Masc=Gen
bil=mEN dAxil hO
burrow.Sg.Fem=Loc enter be.Pres.Sg
gyA
go.Perf.Sg.Masc

'The snake entered the mouse's burrow'

(31) bijlI aor gEs=kE
electricity.Nom.Sg.Fem and gas.Nom.Sg.Fem=Gen
bil har mahInE
bill.Nom.Pl.Masc every month.Nom.Sg.Masc
AtE hEn
come.Pres.Pl.Masc be.Pres.Pl

'Electricity and gas bills come every month'

(32) cUhE zamIn=kE andar
mouse.Nom.Pl.Masc ground.Sg.Fem=Gen inside
gHar banAtE hEn
house.Nom.Sg.Masc make.Pres.Pl.Masc be.Pres.Pl

'Mice build houses underground'

(33) Aj-kal bijlI aor
nowadays electricity.Nom.Sg.Fem and
gEs ka
gas.Nom.Sg.Fem=Gen expense.Nom.Sg.Masc
xarcA boht baRH gyA
very increase.Pres.Sg go.Perf.Sg.Masc be.Pres.Sg
hE

'Nowadays the cost of electricity and gas has increased a lot'

The sample corpus has been selected to analyze the semantics of an Urdu word *bil*. The word has a sense of 'bill' which has been borrowed from English. The second meaning of *bil* is 'mouse's burrow' which is quite different from bill. The diacritic symbols are optional in Urdu script and are usually not written. The same lexicographical representation of the word *bil* produces a pronunciation of *bal*, which further has multiple semantics associated with it. However, the sample corpus depicts the meanings of bill and burrow.

In the sample corpus, (30) is about the entrance of a snake into the mouse's burrow which depicts the first meaning of the

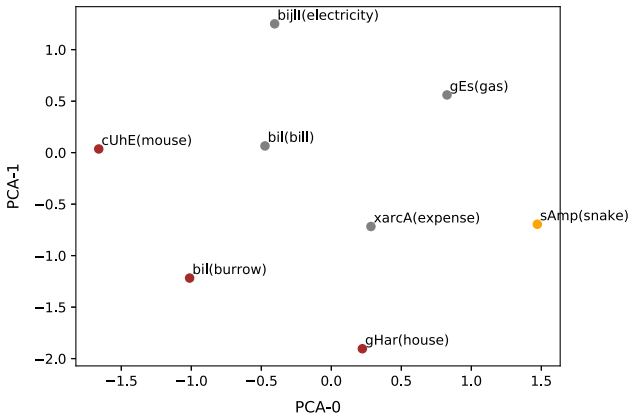

Fig. 5 Feature vector representations of words based on their contexts in the sample corpus

word *bil*. The sentence of (31) presents the monthly billing of electricity and gas which shows the second sense of the same word. The third sentence says about the mice and their underground houses which are referred as burrows. Similarly, the last sentence presents the expense of electricity and gas and its increase which has the sense of billing. We obtained the ELMo vectors for the sample corpus and plotted the feature vectors in a 2-dimensional plane as shown in Fig. 5.

Figure 5 shows the representations of selective words from the sample corpus. The ELMo embeddings have been trained for 128 dimensions with a vocabulary size of 72 thousands. The dimensionality reduction has been performed by using principle component analysis (PCA). It is quite clear that the word *bil* has two different meanings 'burrow' and 'bill.' The words with similar semantics have same color in the graph. The words *cUhE* 'mouse,' *bil* 'burrow' and *gHar* 'house' are represented by brown dots. The words *bijlI* 'electricity,' *gEs* 'gas,' *bil* 'bill' and *xarcA* 'expense' are represented with gray color. Similarly, the word *sAmp* 'snake' is shown by orange dot. The ELMo vectors produce the correct semantics of these words making clusters in a 2-dimensional plane. The word *bil* 'burrow' is near to *cUhE* 'mouse' and *gHar* 'house.' The word *bil* 'bill' is near to *bijlI* 'electricity,' *gEs* 'gas' and *xarcA* 'expense.' We have further computed the cosine similarities of four words *sAmp* 'snake,' *cUhE* 'mouse,' *bijlI* 'electricity' and *gEs* 'gas' in comparison with both representations of *bil* as shown in Table 9.

Table 9 presents the comparison of contextualized ELMo vectors with context-free Word2Vec vectors by computing cosine similarities. ELMo vectors for both senses of the word *bil* are quite visible. The word *bil* 'burrow' has higher similarities with *sAmp* 'snake' and *cUhE* 'mouse' as compared to *bijlI* 'electricity' and *gEs* 'gas.' Similarly, *bil* 'bill' has higher similarities with words *bijlI* 'electricity' and *gEs* 'gas.' On the other hands, the Word2Vec has static word representations and computes a word with single meaning, hence showing uniform cosine similarities with all four words because it did not learn any of these two senses rather some other meanings of the same word due to lexicographic similarity. All the experiments have been performed on a core i7 system with GeForce GTX 1080 Ti graphical processing unit.

6 Results and Discussion

We have performed the chunking experiments by using different feature vectors. The chunking results have been evaluated against f-measures including recall, precision and f-score. The details of f-measures are shown by Equations 3, 4 and 5.

$$Recall(R) = \frac{\#\,Correct\ chunks\ in\ candidate\ set}{All\ chunkes\ in\ gold\ set} \tag{3}$$

$$Precision(P) = \frac{\#\,Correct\ chunks\ in\ candidate\ set}{All\ chunkes\ in\ candidate\ set} \tag{4}$$

$$F - Score(F_1) = \frac{2.P.R}{P + R} \tag{5}$$

Table 10 presents the chunking results against different models which have been trained on the annotated corpus. The experiments have been performed by using the train and evaluation division shown in Table 8. Figure 6 shows training and validation accuracies and loss during the training process by using contextualized word representations. The optimal number of epochs can be achieved by looking at the accuracy and loss curves. The graphs show that the optimal number of epochs is 15 where the loss value is minimum and accuracy is highest for validation set. After this number, model started

Table 9 Cosine similarity of word vectors achieved from ELMo and Word2Vec representations for the sample corpus

Word Vectors	ELMo bil (burrow)	bil (bill)	Word2Vec bil (burrow)	bil (bill)
sAmp (snake)	**0.52666583**	0.2862459	0.25431877	0.2543188
cUhE (mouse)	**0.59819676**	0.2472075	0.27789906	0.2778991
bijlI (electricity)	0.31564699	**0.4458409**	0.21536207	0.2153621
gEs (gas)	0.44589125	**0.4954937**	0.22253805	0.2225381

Table 10 Results of neural phrase chunking using different features

Sr#	Model+Features	Recall	Precision	F-Score
1	1 BiLSTM	89.0	88.5	88.8
2	1 BiLSTM + pos	93.3	92.6	93.0
3	1 BiLSTM + W2V_emb + pos	93.4	93.5	93.4
4	2 BiLSTM + W2V_emb + pos	93.8	93.9	**93.9**
5	3 BiLSTM + W2V_emb + pos	94.2	93.4	93.8
6	1 BiLSTM + ELMo_emb + pos	94.5	94.8	94.6
7	2 BiLSTM + ELMo_emb + pos	94.5	95.0	94.8
8	3 BiLSTM + ELMo_emb + pos	95.1	94.8	**94.9**

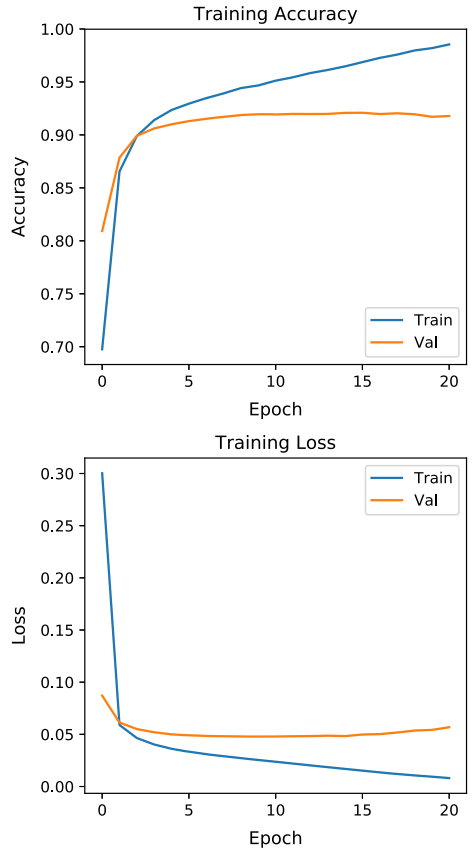

Fig. 6 Accuracy and loss with respect to epochs during the training process

over-fitting. The graph shows the labeling accuracy during the training process; however, the results in Table 10 are presented against phrase chunks.

The first model has been trained by using single LSTM (200 hidden units) layer without merging the details of POS tags and the model performed with an f-score of 88.8. This model took the vector encodings of tokens and learned the embeddings without pre-trained word representations. In the second experiment, we merged the POS vector encodings with the token embeddings, and the chunking results are improved significantly. By incorporating POS tags, the f-score improved by 4.1 points which shows that the POS information is crucial for phrase chunking. Therefore, later experiments have been performed by incorporating POS tag encodings with tokens.

Table 11 Phrasewise chunking results

Sr#	Phrase Labels	Recall	Precision	F-Score
1	NP	91.7	91.8	91.7
2	PP	99.1	98.7	98.9
3	VP	96.4	96.3	96.4
4	O	94.8	93.2	94.0

We further performed transfer learning by incorporating word representations from context-free Word2Vec and contextualized ELMo embeddings. The third model used the trained word embeddings against training tokens along with POS encodings and the chunking results were improved from 93.0 to 93.4. The model was updated to train by using two and three LSTM layers, and the best achieved f-score is 93.9 with two LSTM layers when trained with Word2Vec embeddings.

The ELMo word representations are context sensitive based on the surrounding words. The context is important to learn the correct meaning of a specific word in a sentence. The experiments are further carried by incorporating contextualized ELMo embeddings. The single LSTM layer model, along with ELMo embeddings, outperforms the Word2Vec chunking results with an f-score of 94.6. The further experiments have been performed by using two and three hidden LSTM layers and the f-score improved to 94.9 which is quite promising for a morphologically rich language Urdu. The ELMo representations are character based which are also helpful to learn the morphology of the language.

Table 11 shows the chunking results with respect to phrases. The post-positional and verb phrases have quite high f-scores which are 98.9 and 96.4, respectively. The noun phrases have comparatively lower f-score of 91.7. The corpus contains high number of noun phrases with more diverse syntactic constructions as compared to post-positional and verb phrases. The syntactic variation causes the lower scores

of noun phrases. However, the overall chunking results are quite satisfying.

7 Conclusion

This paper presents the development of a neural phrase chunker and an annotated corpus for Urdu. The corpus provides the annotation of four phrases that are noun phrase, postpositional phrase, verb phrase and prepositional phrase. The corpus has quite even coverage of all the phrases except prepositional phrase, because of its rare appearance in the language. The inter-annotator agreement has been performed on ten percent reference corpus, and the chunk agreement score is 94.53 which is quite acceptable. The corpus contains the text from fifteen text genres, and the annotated phrases have quite evenly coverage among text genres. Bi-directional long–short-term memory (BiLSTM) network-based chunker has been developed to learn the phrase labels. The LSTM-based models are quite capable to learn the sequence labels. Context-free and contextualized word representations have been trained on a plain Urdu corpus to perform transfer learning. The analysis shows that the ELMo embeddings are quite capable to learn the context of words within sentences. The embeddings are character based which are further helpful to learn the morphology. The chunking model, by using the ELMo representations, outperforms the context-free Word2Vec embeddings with an f-score of 94.9 which is quite promising for a morphologically rich language Urdu.

Acknowledgements We are grateful to Prof. Miriam Butt, University of Konstanz, Germany, for providing valuable feedback and hardware support for this work.

References

1. Eberhard, D.M.; Simons, G.F.; Fennig, C.D.: Ethnologue: Languages of the World . SIL International (2019)
2. Bögel, T.; Butt, M.; Hautli, A.; Sulger, S.: Developing a Finite-State Morphological Analyzer for Urdu and Hindi. Universität Potsdam (2008)
3. Hussain, S.: Finite-State Morphological Analyzer for Urdu. Unpublished MS thesis, Center for Research in Urdu Language Processing, National University of Computer and Emerging Sciences, Pakistan (2004)
4. Butt, M.: The Structure of Complex Predicates in Urdu. Center for the Study of Language (CSLI) (1995)
5. Butt, M.; Ramchand, G.: Complex Aspectual Structure in Hindi/Urdu. M. Liakata, B. Jensen, D. Maillat, Eds, 1–30 (2001)
6. Khan, T.A.: Spatial Expressions and Case in South Asian Languages. PhD thesis (2009)
7. Butt, M.; King, T.H.: The Status of Case. In: Clause Structure in South Asian Languages, pp. 153–198. Springer (2004)
8. Raza, G.; Ahmed, T.; Butt, M.; King, T.H.: Argument Scrambling within Urdu NPs. Proceedings of LFG11, 461 (2011)
9. Carreras, X.; Marquez, L.: Phrase Recognition by Filtering and Ranking with Perceptrons. Recent advances in natural language processing III: selected papers from RANLP 2003 260, 205 (2004)
10. Etzioni, O.; Banko, M.; Soderland, S.; Weld, D.S.: Open information extraction from the web. Commun. ACM **51**(12), 68–74 (2008)
11. Ahmed, T.; Urooj, S.; Hussain, S.; Mustafa, A.; Parveen, R.; Adeeba, F.; Hautli, A.; Butt, M.: The CLE Urdu POS Tagset. In: LREC 2014, Ninth International Conference on Language Resources and Evaluation, pp. 2920–2925 (2015)
12. Ali, W.; Malik, M.K.; Hussain, S.; Siddiq, S.; Ali, A.: Urdu Noun Phrase Chunking: HMM based approach. In: 2010 International Conference on Educational and Information Technology, vol. 2, pp. 2–494 (2010). IEEE
13. Ali, W.; Hussain, S.: A Hybrid Approach to Urdu Verb Phrase Chunking. In: Proceedings of the Eighth Workshop on Asian Language Resouces, pp. 137–143 (2010)
14. Asopa, S.; Asopa, P.; Mathur, I.; Joshi, N.: Rule based Chunker for Hindi. In: 2016 2nd International Conference on Contemporary Computing and Informatics (IC3I), pp. 442–445 (2016). IEEE
15. Ehsani, R.; Solak, E.; Yıldız, O.T.: Hybrid Chunking for Turkish Combining Morphological and Semantic Features
16. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.; Dean, J.: Distributed Representations of Words and Phrases and their Compositionality. arXiv preprint arXiv:1310.4546 (2013)
17. Park, S.-B.; Zhang, B.-T.: Text Chunking by Combining Hand-crafted Rules and Memory-based Learning. In: Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics, pp. 497–504 (2003)
18. Le Nguyen, M.; Nguyen, H.T.; Nguyen, P.-T.; Ho, T.-B.; Shimazu, A.: An Empirical Study of Vietnamese Noun Phrase Chunking with Discriminative Sequence Models. In: Proceedings of the 7th Workshop on Asian Language Resources (ALR7), pp. 9–16 (2009)
19. Knutsson, O.; Bigert, J.; Kann, V.: A Robust Shallow Parser for Swedish. In: Proceedings of Nodalida, vol. 2003, p. 2003 (2003)
20. Diab, M.; Hacioglu, K.; Jurafsky, D.: Automatic Tagging of Arabic Text: From Raw Text to Base Phrase Chunks. In: Proceedings of HLT-NAACL 2004: Short Papers, pp. 149–152 (2004)
21. Eiselen, R.: South African Language Resources: Phrase Chunking. In: Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16), pp. 689–693 (2016)
22. Sang, E.F.; Buchholz, S.: Introduction to the CoNLL-2000 Shared Task: Chunking. arXiv preprint arXiv:cs/0009008 (2000)
23. Gharaibeh, I.K.: Development of Arabic Noun Phrase Extractor (ANPE). International Journal on Natural Language Computing (IJNLC) Vol **6** (2017)
24. Prathibba, R.; Padma, M.: Shallow parser for Kannada sentences using machine learning approach. Int. J. Comput. Linguistic. Res. **8**(4), 158–170 (2017)
25. Sun, X.; Nan, X.: Chinese Base Phrases Chunking Based on Latent Semi-CRF Model. In: Proceedings of the 6th International Conference on Natural Language Processing and Knowledge Engineering (NLPKE-2010), pp. 1–7 (2010). IEEE
26. Sun, X.; Nan, X.: Chinese Noun Phrases Chunking: A Latent Discriminative Model with Global Features. In: 2011 14th IEEE International Conference on Computational Science and Engineering, pp. 167–172 (2011). IEEE
27. Sarkar, K.; Gayen, V.: Bengali Noun Phrase Chunking Based on Conditional Random Fields. In: 2014 2nd International Conference on Business and Information Management (ICBIM), pp. 148–153 (2014). IEEE
28. Pawar, S.; Ramrakhiyani, N.; Palshikar, G.; Bhattacharyya, P.; Hingmire, S.: Noun Phrase Chunking for Marathi using Distant Supervision. In: Proceedings of the 12th International Conference on Natural Language Processing, pp. 29–38 (2015)

29. Sassano, M.; Kurohashi, S.: A Unified Single Scan Algorithm for Japanese Base Phrase Chunking and Dependency Parsing. In: Proceedings of the ACL-IJCNLP 2009 Conference Short Papers, pp. 49–52 (2009)
30. Supnithi, T.; Onman, C.; Porkaew, P.; Ruangrajitpakorn, T.; Trakultaweekoon, K.; Kawtrakul, A.: A Supervised Learning based Chunking in Thai using Categorial Grammar. In: Proceedings of the Eighth Workshop on Asian Language Resouces, pp. 129–136 (2010)
31. Nongmeikapam, K.; Chingangbam, C.; Keisham, N.; Varte, B.; Bandopadhyay, S.: Chunking in Manipuri using CRF. Int. J. Nat. Lang. Comput. (IJNLC) **3**(3) (2014)
32. Aung, M.P.; Moe, A.L.: New phrase chunking algorithm for Myanmar natural language processing. In: Applied Mechanics and Materials, vol. 695, pp. 548–552 (2015). Trans Tech Publications
33. Ehsan, T.; Hussain, S.: Development and evaluation of an Urdu treebank (CLE-UTB) and a statistical parser. Language Resourc. Eval. 1–40 (2020)
34. Ehsan, T.; Hussain, S.: Analysis of experiments on statistical and neural parsing for a morphologically rich and free word order language Urdu. IEEE Access **7**, 161776–161793 (2019)
35. Ahmed, T.; Ehsan, T.; Ashraf, A.; u Rahman, M.; Hussain, S.; Butt, M.: A Multilayered Urdu Treebank. In: International Conference on Language and Technology (CLT 2020) (2020)
36. Ehsan, T.; Butt, M.: Dependency parsing for Urdu: resources, conversions and learning. In: Proceedings of The 12th Language Resources and Evaluation Conference, pp. 5202–5207 (2020)
37. Kamran Malik, M.; Ahmed, T.; Sulger, S.; Bögel, T.; Gulzar, A.; Raza, G.; Hussain, S.; Butt, M.: Transliterating Urdu for a Broad-Coverage Urdu/Hindi LFG Grammar. In: LREC 2010, Seventh International Conference on Language Resources and Evaluation, pp. 2921–2927 (2010)
38. Jespersen, O.: A Modern English Grammar on Historical Principles, vol. 3. Routledge (2013)
39. Gómez, I.P.: Nominal Modifiers in Noun Phrase Structure: Evidence from Contemporary English. University of Santiago de Compostela (2010)
40. Bharati, A.; Sangal, R.; Sharma, D.M.; Bai, L.: Anncorra: Annotating Corpora Guidelines for POS and Chunk Annotation for Indian Languages. LTRC-TR31, 1–38 (2006)
41. Bhatt, R.; Farudi, A.; Rambow, O.: Hindi-Urdu Phrase Structure Annotation Guidelines (2013)
42. Anwar, B.: Urdu-English code switching: the use of Urdu phrases and clauses in Pakistani English (a non-native variety). Int J Language Stud **3**(4) (2009)
43. Cohen, J.: A coefficient of agreement for nominal scales. Educ. Psychol. Measur. **20**(1), 37–46 (1960)
44. Adeeba, F.; Akram, Q.; Khalid, H.; Hussain, S.: Cle Urdu Books N-Grams. In: Conference on Language and Technology (2014)
45. Pennington, J.; Socher, R.; Manning, C.D.: GloVe: global vectors for word representation. In: Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543 (2014). http://www.aclweb.org/anthology/D14-1162
46. Peters, M.E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; Zettlemoyer, L.: Deep contextualized word representations. arXiv preprint arXiv:1802.05365 (2018)