Artif Intell Rev (2017) 47:279–311
DOI 10.1007/s10462-016-9482-x

Urdu language processing: a survey

Ali Daud[1] · **Wahab Khan**[1] · **Dunren Che**[2]

Published online: 2 June 2016
© Springer Science+Business Media Dordrecht 2016

Abstract Extensive work has been done on different activities of natural language processing for Western languages as compared to its Eastern counterparts particularly South Asian Languages. Western languages are termed as resource-rich languages. Core linguistic resources e.g. corpora, WordNet, dictionaries, gazetteers and associated tools being developed for Western languages are customarily available. Most South Asian Languages are low resource languages e.g. Urdu is a South Asian Language, which is among the widely spoken languages of sub-continent. Due to resources scarcity not enough work has been conducted for Urdu. The core objective of this paper is to present a survey regarding different linguistic resources that exist for Urdu language processing, to highlight different tasks in Urdu language processing and to discuss different state of the art available techniques. Conclusively, this paper attempts to describe in detail the recent increase in interest and progress made in Urdu language processing research. Initially, the available datasets for Urdu language are discussed. Characteristic, resource sharing between Hindi and Urdu, orthography, and morphology of Urdu language are provided. The aspects of the pre-processing activities such as stop words removal, Diacritics removal, Normalization and Stemming are illustrated. A review of state of the art research for the tasks such as Tokenization, Sentence Boundary Detection, Part of Speech tagging, Named Entity Recognition, Parsing and development of WordNet tasks are discussed. In addition, impact of ULP on application areas, such as, Information Retrieval, Classification and plagiarism detection is investigated. Finally, open issues and future directions for this new and dynamic area of research are provided. The goal of this paper is to organize the ULP work in a way that it can provide a platform for ULP research activities in future.

✉ Ali Daud
ali.daud@iiu.edu.pk

[1] Department of Computer Science and Software Engineering, IIU, Islamabad 44000, Pakistan

[2] Department of Computer Science, Southern Illinois University, Carbondale, IL 62901, USA

Keywords Urdu language processing (ULP) · Datasets · Characteristics · Natural language processing (NLP) · Part-of-speech (POS) · Named entity recognition (NER) · Sentence boundary detection (SBD)

1 Introduction

In the past few years, multilingual content on the internet increased rapidly. Consequently, monolingual and cross-lingual Information Retrieval (IR) task has gained a lot of attention from the NLP researcher community. WWW was initially a web of English language and later on became multilingual. Monolingual IR is focused on the queries and information accessed in the same language, while cross-lingual IR is focused on the queries and information accessed in several different languages (Capstick et al. 2000).

Indian and similar languages attracted researcher's attention during recent years. Especially, Urdu language started to become a major part of Asian languages on web (Mukund et al. 2010).

IR and Data Mining (DM) tasks, such as, relationship exploration, topic categorization, event extraction, sentiment analysis involves detailed knowledge of NLP. Importance of NLP tasks; stop words removal, parsing, POS tagging, morphological analysis, shallow parsing and NER have significant importance in all NLP systems (Riaz 2010). NLP systems for English are quite mature but Urdu NLP systems needs a lot of effort to be made yet (Al-Shammari 2008; Jawaid and Ahmed 2009; Adeeba and Hussain 2011).

National Language of Pakistan is Urdu. It is among the most spoken languages in India. Approximately, there are 11 million speakers of Urdu are in Pakistan and 300 million plus in the whole world (Riaz 2008a). Pakistan, India, USA, UK, Canada and USA have Urdu speakers in abundance. The Urdu language family tree can be described as: Indo-European→Indo–Iranian→Indo-Aryan→Urdu (Humayoun et al. 2007). Recently, computational processing of the languages with script writing style from right to left, e.g. Urdu, Arabic got significant attention of NLP researchers. Especially Arabic is a semitic language and has been investigated intensely. Dari, Punjabi, Pashto and Persian (Farsi) belong to Proto Indo Iranian languages. They also follow right to left script writing style and are widely spoken in South Asia region. These languages have some writing and speaking similarities but needs individual attention for most tasks e.g. a stemming technique for one language might not work well for other language (Riaz 2007, 2010).

Urdu has its roots in Persian, Arabic and similarities with most South Asian languages. For example, similarity in terms of: lack of capitalization, lack of small and capital words and free word order characteristic. Urdu have structural similarity with Hindi (Ahmed and Hautli 2010; Visweswariah et al. 2010). The dissimilarity is in writing style and vocabulary e.g. Devanagri script is used for writing Hindi whereas Urdu is written in Perso-Arabic script (Riaz 2010). Urdu is comparatively complex as its morphology and syntax structure is a combination of Persian, Sanskrit, English, Turkish and Arabic (Adeeba and Hussain 2011).

Previously, not much work is done about ULP due to little attention of language engineering community and less availability of linguistic resources. A few survey papers have been written on Urdu and its related issues. But all of them have focused on one or two tasks of ULP (Anwar et al. 2006; Riaz 2008a). Riaz (2008b) discussed new techniques for tasks like Stop Words Identification, Stemming, Concept Searching and NER. In this survey, we have tried to cover most details of datasets, characteristics, tasks and techniques available for ULP as well as its application areas. The contributions of this paper are:

1. Highlighting the importance of ULP
2. Providence of the available data resources for ULP
3. Description of characteristic, morphology and resource sharing between Hindi and Urdu
4. Categorization of tasks in ULP
5. Classification of techniques which are available to deal with these tasks
6. Insights about application areas of ULP
7. Intuitive future directions

This paper is especially helpful for new researchers to find most information about ULP at one place in a compact way.

The remaining part of the survey is organized as follows. Section 2 provides details of the available datasets. In Sect. 3, general characteristics of Urdu language and the differences between Urdu and Hindi are discussed. In Sect. 4, the pre-processing on Urdu data, such as, stop word removal, text normalization and stemming is discussed. In Sects. 5 and 6, different tasks regarding ULP and classification of techniques and their summary is provided and Sect. 7 explores impact of ULP on IR, classification and plagiarism detection. Section 8 provides future directions and finally Sect. 9 concludes.

2 Datasets

Gold-standard data is extremely important for NLP techniques, whether statistical or rule-based. In order to train various models or algorithms effectively, it's important to provide right training data to models and also the data must be large enough to train model properly (Riaz 2010).

Now a days the two natural choices that are widely considered during the creation phase of datasets for Arabic script based languages are: (a) to use Unicode character set for storing the data (b) to store the corpus data in XML file format. Urdu an Arabic script based language also uses Unicode encoding scheme for storage purpose. Although, Urdu text can be stored in multiple file formats e.g. in simple ".txt" or ".doc" etc but naturally XML is most widely adoptable file format system for Urdu dataset storage (Becker and Riaz 2002). The main advantage of using XML file format system for data storage purpose includes: (a) Data stored in this format can easily converted to other formats automatically and (b) The document is readable by both computers as well as technically aware humans (Henderson and Deane 2003).

There is need for large datasets for performing different natural language tasks and applications. Consequently, a large dataset is used by Ali and Ijaz (2009) for classification task. It was consisted of 19.3 million words with documents classified in six categories namely; finance, culture, sports, news, personal and consumer information. They preprocessed the text of six domains by performing tokenization, diacritics elimination, normalization, Stop word removal, stemming and also using some statistical techniques. After preprocessing the datasets were standardized and their size was reduced. For example, before preprocessing the news domain was consisted of 78,649 word types and after preprocessing the vocabulary was reduced to 54,817 word types. The authors referred the word types in the preprocessed domain or class as Terms. The class wise input dataset development process details are given in Table 1. In total, there were 26,067 documents. The white spaces and punctuation marks are used for tokenization. The tokenization lexicon is manually prepared after gathering data from different sources which contains 220,760 unique tokens. All the words are matched in the tokenization lexicon if found it becomes a token otherwise the word is ignored (Fig. 1).

Table 1 Preprocessing analysis of documents (Ali and Ijaz 2009)

Class	Documents	Tokens	Types	Terms
News	17,501	8,957,259	78,649	54,817
Sports	3388	1,666,304	21,473	16,622
Finance	1766	1,162,019	16,144	11,951
Culture	1088	3,845,117	57,486	37,493
Consumer information	1046	1,980,723	26,433	19,781
Personal communication	1278	1,685,424	34,614	25,588
Total	26,067	19,296,846	234,799	166,252

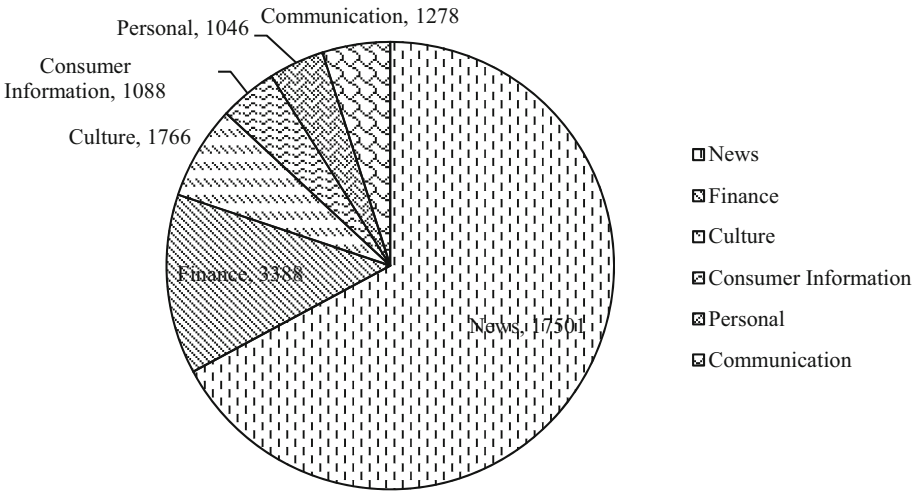

Fig. 1 Graphical representation of six classes' data (Ali and Ijaz 2009)

Unfortunately, there is not a good number of datasets are available for doing ULP research and developing tools. The commonly used datasets for various ULP tasks are given below. There are two ULP datasets about which discussion can be found in literature, but currently they are unavailable from their source URLs are CRULP (Centre for Research in ULP) and CRL (Computing Research Laboratory).

2.1 Becker–Riaz dataset

Becker–Riaz Urdu dataset (2002) is the first Urdu linguistic resource that is made publically available in 2002 for conducting research in ULP domain. Unicode character using Arabic script set is used to store Urdu text. It is marked up according to Corpus Encoding Standard (CES) and XML Document Type Definition (DTD) with metadata and tags in English (Becker and Riaz 2002). It contains 7000 short news articles collected from BBC news and has a very rich content for NER.

2.2 The EMILLE dataset

The commonly used dataset for ULP released by Lancaster University in 2003 is the EMILLE (Enabling Minority Language Engineering). The objective was to construct a 67 million word dataset of South Asian languages (Baker et al. 2003). But after regular additions with the passage of time, now the EMILLE monolingual dataset contains more than 96 million words. It is based on CES and is in XML format. Its three major constituents were Monolingual, Parallel and annotated data. The monolingual dataset comprises of both written and spoken corpora. The Urdu corpus that was made available by EMILLE project for research and development consists of 512,000 words of spoken Urdu and 1,640,000 words of Urdu text. Besides the language of Urdu, written corpora of the EMILLE project is also based on monolingual corpora for thirteen South Asian languages (Hardie 2003).

Another English dataset of EMILLE project contains 200,000 words with its supplementary translation in Urdu, Gujarati, Punjabi, Hindi and Bengali. It also provides annotated data with Urdu written, spoken and parallel corpora annotated with a large morpho-syntactic tag-set through Urdu tagger.

2.3 CLE dataset

The Center for Language Engineering in Pakistan has also taken initiatives efforts in corpus-building activities. Center for Language Engineering (CLE) for research and computational processing in Urdu launched Urdu Digest POS Tagged dataset taken from Urdu digest between 2013–2011 (CLE 2015). It contains 100 K Urdu words from several domains, such as, politics, health, education, international affairs, business, humors, sports and literature. It has two major categories (1) The Informational (80 %) and (2) Imaginative (20 %). The Informational part includes text from politics, health, education, international affairs, entertainment, and science while imaginative part includes texts from book reviews, novels, translation of foreign literature and short stories. The only limitation of CLE POS tagged dataset is its license restriction.

2.4 IJCNLP-2008 NE tagged dataset

The IJCNLP-2008 NE tagged Dataset is a corpus of about 40,000 words annotated with twelve named entity classes. Center for Research in Urdu Language Processing (CRULP) at National University of Computer and Emerging Sciences in Pakistan and IIIT Hyderabad, India jointly created this annotated corpus and donated to the NER workshop (Hussain 2008). This corpus is freely available from the given URL http://ltrc.iiit.ac.in/ner-ssea-08/index.cgi?topic=5 (Table 2).

3 Characteristics

The most important characteristic of Arabic script languages is context sensitivity, where the letters change their shape with respect to their next letters. Urdu is inflectional rich language that extracts its own vocabulary from many other languages, such as, from Persian and Arabic (Hardie 2003). Urdu also borrows its vocabulary from Turkish, Portuguese and English (Anwar et al. 2006; Akram et al. 2009; Adeeba and Hussain 2011; Ahmed and Hautli 2011). It is the combination of many languages along with its own morphology and is also influenced by the morphology of Farsi language (Anwar et al. 2007; Becker and Riaz 2002;

Table 2 Summary of ULP datasets

Dataset name	Release year	Task	No. of words	No. of documents
Becker–Riaz dataset	2002	NER, stop words removal, IR, stemming, base line evaluation and comparison with Hindi and Urdu	20,000–50,000	7000
EMILLE dataset	2003	Stop words removal, segmentation, NER, translating, POS tagging, anaphoric annotation, language engineering analysis	1,640,000 (Urdu written) 512,000 (Urdu spoken)	300 (Text documents)
IJCNLP-2008 NE tagged dataset	2008	NER	40,000	Training: 5 documents Testing: 1 document
CLE dataset	2012	POS tagging, NER	100,000	348

Table 3 Example of similar meanings sentences

بنایا	شناختی کارڈ	نیا	نے	عنزہ
Banaya	Shanakhti Card	Naya	Nay	Anzah
Anzah made a new ID Card				
بنایا	نے	عنزہ	شناختی کارد	نیا
Banaya	Nay	Anzah	Shanakhti Card	Naya
Anzah made a new ID Card				

Riaz 2007). NLP for Urdu is important due to its unique nature, morphology and also due to its millions of speakers around the globe. Due to its rich morphology and exceedingly inflected nature, it is always remained a favorite and preferable choice of poetry writers and that's why it is also termed as a language of poetry (Riaz 2010, 2007; Naz et al. 2012). Besides its huge significance, it does not have abundant linguistic resources for performing various intellectual ULP tasks (Riaz 2010; Durrani and Hussain 2010).

In Urdu, we have more than one word order for similar meaning sentences; therefore Urdu is also termed as free word order language (Riaz 2010). Table 3 explores free word order characteristic of Urdu language where both the sentences represent the same concept.

Spelling Variations, Ambiguity in Suffixes, Loan words, Nested Entities, Conjunction Ambiguity and Resource Challenges are main challenges in Urdu text processing (Riaz 2010; Singh et al. 2012).

Urdu is considered as the lingua franca of business in Pakistan, and the South Asian community of UK (Riaz 2010). Urdu and Hindi are closely related languages; a claim can be made that any computational model or algorithm that works for Hindi might work for Urdu as well. Hindi and Urdu are two institutionalized types of Hindustani, an Indo-Aryan dialect that holds a few nearly related dialects of northern India and Pakistan. Urdu is recognized to be very much like Hindi as it imparts its phonological, morphological, and syntactic structure with Hindi. Both these dialects are developed from Sanskrit and offer the regular Khari Boli vernacular. They are free word order dialects and follow a general SOV (Subject-Object-Verb) structure. Anyhow in spite of the resemblance NLP tools created for Hindi can't be utilized for Urdu (Mukund et al. 2010; Riaz 2010, 2012). The three main differences pointed out by Riaz (2012), Mukund et al. (2010), due to which tool developed for Hindi NLP task can't be used directly for carrying out Urdu NLP tasks are given below:

a. **Script difference:** Urdu script is predominantly written in Nastaliq style; similar to Arabic script based languages, in which text is written from right to left. Hindi on the other hand, is written in Devanagari script which flows from left to right.
b. **Vocabulary difference:** The Urdu vocabulary influenced majorly by Persian, Turkish and Arabic, while Hindi vocabulary is influenced by Sanskrit.
c. **Missing diacritics problem:** The Hindi tools which consider token level features cant be directly used due to missing diacritics problem in Urdu.

Despite these dissimilarities, both can be treated as same language at high level. This can play vital part in developing links among other South Asian communities across the world. The two languages share many morphological and syntactic characteristics but for NLP applications there is the issue that the scripts and many vocabulary words are different. Conclusively, both languages need individual attention of researchers (Riaz 2010). Some resources can be shared or modified by using transliteration schemes and adaptations for the vocabulary differences. We will address this potential sharing of resources in more detail in the following sections.

3.1 Resources sharing between Hindi and Urdu

Most of the worlds related languages share their advanced vocabulary with each other although they have variation in their basics. Hindi and Urdu share a common phonological, morphological and grammatical structure but script writing style of both are different. In addition, the vocabularies have also diverged significantly especially in the written form (Visweswariah et al. 2010; Riaz 2012). In Riaz (2008b, 2009, 2012). The author argued for independent innovative work for the Urdu language as opposed to depending on tools and resources developed for Hindi language. Urdu, the national language of Pakistan and Hindi, the national language of India, share every day basic vocabulary, such that speakers of both these languages can understand each other as if they were isotopes of a mutual language. Although both languages share basic vocabulary of every day speech and grammar (Riaz 2009; Flagship 2012; Riaz 2012), but they are often mutually incomprehensible. In order to demonstrate that Hindi and Urdu have similarity in their grammatical structure as well as in basic vocabulary (Prasad and Virk 2012) reported computational translation evidence of this unusual relationship between Urdu and Hindi. They took Grammatical Framework (GF) (Ranta 2004), a grammar formalism tool commonly incorporated for development of multilingual grammars which can latterly be used for tasks such as translation. They modified GF mechanically for Hindi, by incorporating necessary changes only in the script and lexicon where needed. During assessment phase, the Urdu grammar and its Hindi twin either both effectively decoded an English sentence, or were unable to translate in exactly the same grammatical way. The results computationally confirmed that Hindi and Urdu share a grammar but differ so much in vocabulary that they should be treated as two separate languages in any circumstances, except in some most basic situation. Riaz (2008a, 2009, 2012) discussed that the existing linguistic resources and available technologies for Hindi can't be used as bridge to conduct research in ULP, but for some tasks such as statistical machine translation, Hindi POS and Hindi-English word aligner Visweswariah et al. (2010) showed that the two languages resources are inter operable.

Visweswariah et al. (2010) showed that even in the absence of parallel corpus a good quality translation system between Hindi and Urdu can be resulted. They incorporated translation techniques in order to share linguistic resources (Hindi-English parallel corpus, Urdu POS corpus and manually word aligned Urdu-English corpus) between the two languages. They demonstrated improvements on three tasks (1) they reported improvement up to 0.8 in BLEU score for the task of statistical machine translation from Urdu to English by using Hindi-English parallel corpus (2) they reported improvement in Hindi POS tagging up to 6 % by using an Urdu POS corpus and (3) they showed improvement in Hindi-English word aligner up to 9 % absolute in F-Measure by using a manually word aligned Urdu-English corpus.

3.2 Urdu orthography

The script of Urdu is traditionally and predominantly written in Nastaliq style that makes it more challenging among the languages which follows Arabic script. As compared to other existing scripts, the distinguishing feature of Arabic script exists in its writing style. Text is written from right to left in all those languages which follows Arabic script. Arabic script based languages are context sensitive and due to this context sensitive characteristic these languages are written in the form of ligatures, the resulted ligatures might consist of a single graphemes/character or union of several characters to form a word. Important characteristics of Nastaliq Style is described in the below text. In Nastaliq style most of the characters acquires different shapes depending on their position in the ligature e.g. a letter

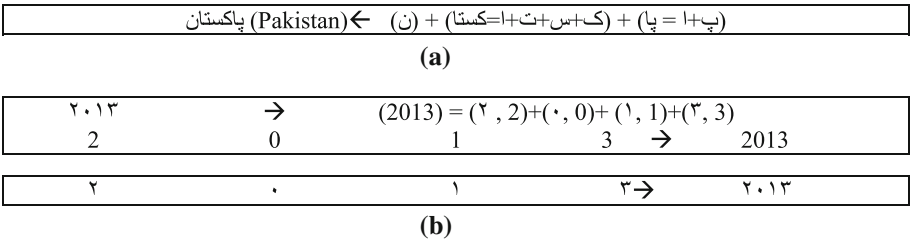

Fig. 2 **a** Nastaliq character writing architecture. **b** Nastaliq digits writing architecture

may appear differently depending on its position as an isolated, middle, center, or ending character (Imran 2011). Nastaliq has vertical stacking of characters as they are kerned and cursively joined while some characters move backward and beyond the previous character (Sattar 2009).

Basically Nastaliq style follow two dimensional architecture where characters are written from right to left shown in Fig. 2a and the digits follow writing style from left to right shown in Fig. 2b where '۲', '۳' and '۰' represents digits whose English equivalent digits are '2', '3' and '0', respectively (Adeeba and Hussain 2011).

3.3 Urdu morphology

Urdu is a morphologically rich language, which means in Urdu it is possible that for a single word there exist various variants. In NLP, morphology plays an important role. Morphology is the study of word structure (Gupta et al. 2015). For morphology, Urdu exhibits the characteristics of other Indo-European languages, e.g. having concatenative inflective morphological framework.

3.3.1 Nouns

Urdu noun اسم *(Ism)* has two grammatical Genders: masculine مذکر and feminine مؤنث . Nouns may have particular gender suffixes (marking), or be unmarked for gender. Nouns are inflected to show case and number (singular or plural). Marking subdivides Urdu nouns into two groups: (a) all those nouns which have clear gender suffix are known as marked nouns and (b) all those nouns which have no special gender suffix are known as unmarked nouns. Detail is as follows:

The gender of many nouns could be known by their terminations. Nouns ending with suffixes "ا" (*a*), ہ (*ha*) and یہ (*yaa*) are masculine. e.g. لڑکا (*Larka,* Boy), مرغا (*Murgha,* Rooster), بچہ (*Bacha*, Male child) and روپیہ (*Rupayaa,* Money). But in some exceptional cases these rules are declined, because in Urdu some words are by default feminine, although these words have masculine gender suffix e.g. ملکہ (Malkah, Queen), انّا (Ana, Ego), تنخواہ (Tankhwa, Salary).

Nouns ending with suffixes "ی" (i) or "یا" (ya) are feminine. e.g. لڑکی (*Larki,* girl) مرغی (*Murghi,* Hen), بچی (*Bachi*, Female child) and چڑیا (*Chiriya,* Bird).

Nouns that have no special gender ending suffix are termed as unmarked nouns, and their gender is determined through careful learning procedure. Examples of unmarked masculine and feminine nouns are: گھر (Ghar, House) (masculine), کام (Kaam, Work)(masculine), کتاب (Kitaab, Book)(Feminine).

Nouns that are inflected to show number have different plural suffixes for masculine and feminine nouns. The suffix "ا" (*a*) changes to "ے "(*ay*) e.g. لڑکا (Larka, Boy) → لڑکے (Larkay,

boys), بچہ (Bacha, Child) → بچے (Bachay, Children). The suffix " یہ " (*yaa*) changes to "ے" (*ay*) e.g. روپیہ (*Rupayaa,* Rupee)→ روپے (*Rupay, x*Rupees). The suffix "ی" (*i*) changes to "یاں" (*iyan*) e.g. لڑکی (*Larki*, Girl)→ لڑکیاں (*Ladkiyan*, girls). The indigenous feminine nouns ending in " یا " (ya) takes the plural in "یاں" (*iyan*). e.g. چڑیا (*Chirya,* Bird)→ چڑیاں (*Chiriyan*, Birds). The feminine unmarked noun adds the plural suffix "یں" (ain). e.g. کتاب (*kitab,* Book)→ کتابیں (*kitabain,* Books) (Small and George 1908; Schmidt 1999).

Noun may occur in Nominative, Oblique or vocative case. Nominative nouns most commonly occur as the subject of the verb. e.g:

لڑکا یہاں رہتا ہے۔
hay rehta yahan *Larka*
The *boy* lives here.

Whenever, a noun is followed by a postposition (for example ko کو "to"; ka, کا "of"; main میں "in"; se سے "from"), it occurs in the oblique case. The example of oblique singular and plural is given below.

Oblique Singular Example:
لڑکے کا والد کراچی میں ہے۔
hay main Karachi walid ka *Larkay*
The *boy's* father is in Karachi.

Oblique Plural Example:
لڑکیوں کا والد کراچی میں ہے۔
Hai Main Karachi walid Ka *Larkiyon*
The *girls'* father is in Karachi.
The father of the girls is in Karachi.

The vocative is used only toward persons or objects identified with persons, and do not occur very often. Vocatives may be introduced by the vocative interjections. e.g.او(o), اے (ay) or ارے (aray) (Small and George 1908; Schmidt 1999).

بیٹے، ادھر آو۔
aao idhar Baity
Son, come here.

او رکشے والے!
wale rikshay O
O rikshaw driver!

3.3.2 Verbs

In most languages the verb stem refers to the base morpheme that indicates the meaning of the verb. It is the base form of the verb and can usually take affixes (suffixes and prefixes). In English for example, the word "run" is the verb stem. To this stem "run" we can add the suffix "ing" to indicate present continuous aspect. In other words, verb stem is the basic verb form that usually exists in dictionaries. In some languages like Urdu it can be quite complex.

Verb (فعل , Fael) corresponds to occurrence or performing some action. That verb which does not take object is called intransitive verb (فعل لازم , Fael Lazim). When a verb needs a direct object then it is called transitive verb (فعل معتدی Fael Muatadi).

Urdu verbs have four basic forms: the Root (Stem), Imperfective participle (اسم حالیہ , *ism-i-Halia*), perfective participle (اسم مفعول , *ism-i-Maful*), and infinitive (مصدراسم , *Masdar*) (Schmidt 1999).

- **Infinitive:** The infinitive (مصدر, Masdar) of the verb is that part which is given in the dictionaries. The root, the present and past participle are derived from this form. These are the principle part of verb. Three tenses are formed from each of these, making in all nine principle tenses of the verb. The infinitive form of a verb can be used in place of nouns, as a request form and in infinitival constructions showing necessity, advisability, obligation, imminence, the agent, permission, purpose and negative assertion (Schmidt 1999).

The infinitive form of a verb has the following properties:

- It always represents some action
- It always ends with infinitival suffix "نا" (*na*) and may inflected masculine nouns. e.g. "کرنا" (Karna, to do, to act) and "سننا"(Sunna, to hear, to listen),"کھیلنا" (Khelna, to play).
- It does not need any tense for delivering its meaning.
- When the suffix "نا" (na) is taken away from the infinitive, what is left is called the root of the verb, which is also called (مادہ) (Madah, Feminine) in Urdu.
- **Root**: A root form is a morpheme of Urdu verb which does not change among different morphological forms and is also called base form. The root or stem is the second person singular of the imperative, and is derived from the infinitive by cutting of the termination "نا" *na;* as from "بولنا" (*Bolna*,Talking)→ "بول" (*Bol,*Talk), from "جانا" *Jana*, to go) → "جا" (*Ja*, go) and from "سننا" *Sunna*, to listen) → "سن" (*Sunn*, Listen).
- **The Imperfective or Present Participle:** The Imperfective or Present Participle (اسم حالیہ, *ism-i-Halia*) is formed from infinitive by changing the infinitive suffix "نا"*na* to present suffix "تا" ta; as from "بولنا" Bolna, to Talk)→ "بولتا" (Bolta, talking) and from "کرنا" Karna, to do) →"کرتا" (Karta, doing). In some exceptional cases this rule will not work. Further details can be found in Schmidt (1999).
- **Perfective Participle:** The perfective participle is formed from the root by the addition of past suffix "ا" (a), which is inflected like an adjective to agree with nouns or pronouns in gender and number. e.g. "کرانا" (Karana, to cause to be done)→ "کرایا" (karaya, caused to be done).

4 Pre-processing

Data pre-processing in any language is an important step carried out before applying NLP, IR and DM techniques. This module consists of four subtasks: stop words removal, diacritics removal, Normalization and Stemming.

4.1 Stop words (conjunction words) حرف جار (Haroof Jar)

Natural language is composed of two types of words: content words that have meaning associated with them and functional words that don't have any meaning. Stop words also called negative list and is used to identify function words that don't need to be indexed because no one uses them as a query word (Riaz 2008a). Stop words are functional words

of a language and meaningless in context of text classification. They are eliminated from the lexicon in order to reduce its size by using a list of most frequent words known as stop word list. Stop words are those words that if used in a query will return a large number of documents, possibly the whole corpus. If too many documents are returned, then no IR is accomplished (Riaz 2007). Haroof Jar (Conjunction words) are filtered before performing any IR, DM or classification task because these words don't need to be indexed and therefore can save disk requirements as well as better results can be obtained (Riaz 2008a). (That, woh,وہ), (This, yeh, یہ),(And, Aur, اور) and (Ka, کا Ki, کی) are the examples of Haroof jar. In Urdu text classification, Ali and Ijaz (2009) have compared the result with and without removing "Haroof Jar" to show the effectiveness of performing this pre-processing step.

4.2 Diacritics removal

Diacritics are used in Urdu text to alter pronunciation of a word but they are optional characters in the language. In Urdu, diacritics/Aerab (zer, zabar, and pesh) are not consistently marked and their usage is left to the writer's discretion, creating a one-to-many ambiguity. So missing diacritics from words such as تیر (Teer, Swim) and تِیر (Tir, Arrow) دَم (Dam, strength) اوردُم (Dum, Tail), بَکری (Bakri, Goat), بِکری (Bekri, Daily Transaction) بَندوں (Bandoon, Men) and بِندوں (Bindoun, a type of jewelry) گنا (Ganna, Sugar Cane) and گُنا (Gunna, Number of Times) creates ambiguity which is also termed as ambiguity of Zabar and Zer. The available virtual Urdu keyboards have the option to allow users for including diacritics like zer (Arabic Kasra), zabar (Arabic Fatha) and pesh (Arabic Damma) in text during typing, represented by the Unicode (U+0650) and (U+064E) and (U+064F), respectively. But it is commonly observed that most users avoid diacritics insertion in text during typing except in some exceptional case where its usage is necessary. So a word such as بِندوں (Bindoun, a type of jewelry) which contains *zer* below "ب" are written without *zer* such as "بندوں" . Therefore in such situations diacritics normalization is required to overcome the aforementioned inconsistency in the text. Available choices for Urdu Diacritics removal are: either to restore all the diacritics or completely remove the diacritics. The restoration technique is very complicated because a lot of resources such as Lexicon, annotated corpora as well as other resources in which words having diacritics have to be restored. So the ultimate option for bringing consistency and to standardize the corpus, is the diacritics are completely removed (Ali and Ijaz 2009; Mukund et al. 2010). The Transliteration utility developed by CLE[1] for mapping Urdu Unicode to ASCII encoding have the options either to diacritize the input text before Transliteration or not.

4.3 Text normalization

Text Normalization is the process of converting multiple equivalent representations of data into its standard form in the language. Unicode Normalization standard[2] defines two types of equivalence, canonical equivalence and compatibility equivalence for bringing equivalence between characters and four Unicode normalization forms namely: Normalization Form D (NFD), Normalization Form C (NFC), Normalization Form KD (NFKD) and Normalization Form KC (NFKC). Details of different normalization forms, description of normalization process and summary of Normalization algorithms can be found on the Unicode website.[3]

[1] http://www.cle.org.pk/software/langproc/transliterator_tools.htm.

[2] http://www.unicode.org/reports/tr15/.

[3] http://www.unicode.org/reports/tr15/.

Table 4 Example of verb stemming

Verb	Root Form
جانا (Jana, to go)	جا (Ja, go)
کرنا (Karna, to do)	کر (Kar, do)
دینا (daina, to give)	دے (day, give)
سننا (Sunna, to listen)	سن (Sunn, Listen)

The Urdu Normalization[4] utility provided by CLE is based on the first three normal forms defined by Unicode normalization standard.

Text Normalization is an important and necessary processing step for a wide range of NLP tasks such as IR, text-to-speech synthesis, speech recognition, information extraction, parsing, and machine translation (Zhang et al. 2013).

Urdu Text Normalization is necessary before the aforementioned tasks, because some Urdu alphabets have more than one Unicode as they are shaped similar to Arabic alphabets. For example, character آ has two representations, Unicode value U+0622 and also Unicode values U+0627 + U+0653. Therefore, in order to keep the Unicode of the characters consistent text normalization is carried out. Also in Urdu some characters have different orthographic forms and these variations cause discrepancies in NLP. However, most writers tend to use these variants inconsistently. For example, such as the use of ي vs. ی (Yeh vs. Alif Maqsura (ya)) and the use of ہ vs. ۃ (Heh vs. Taa Marbuta). During text normalization process such characters are replaced by alternate Urdu alphabets to stop creating multiple copies of a word (Ali and Ijaz 2009).

4.4 Stemming

Stemming is another main data pre-processing activity. The objective of stemming is to standardize words by reducing a word into its origin or root (Riaz 2007, 2008b). Stemming is usually performed when dealing with textual data prior to IR, DM, and NLP (Paik et al. 2011; Estahbanati and Javidan 2011). One of the major utility of stemmers is that it is used to enhance the recall of a search engine (Riaz 2008b). Stemming consists of reducing a given word to its stem, base or root, e.g. the stem of لڑکیاں (Larkian, Girls) is لڑکی (Larki, Girl) and the stem of کتابیں (Kitabein, Books) is کتاب (Kitab, Book).

Stem is the basic form of the word that has no inflectional element and produces a root form (Riaz 2007). Morpheme is the unit of language that reflects a meaningful form of the word (Rizvi and Hussain 2005).

Causative Stem Form of verb can be achieved through adding of suffixes to root form. The Causative verb forms / transitivitized verb forms can be obtained through the roots of lower valency verb by adding Urdu suffixes: –aa (ا), –waa (وا) to the root form of verb. The causative verb types are known as stem forms.

Following are some examples of verb stemming (Table 4).

Urdu has a high agglutinative nature in which a word may consist of prefixes, lemma (canonical form of a word) e.g. from the word مردوں (Mardoon, Men) the word مرد (Mard, Man) and from the word گبھرانا (Ghabrana, Fearing) the word گبھراہٹ (Ghabrahat, Fear) and suffixes in different combinations, which results in a very complicated morphology. e.g. Word = prefix(es) + lemma or Stem + suffix(es). e.g. ناک (Naak, Nose) is also a free morpheme ("Nose") and a suffix in word that makes adjective from noun e.g., خطر ناک (KhatarNaak, Dangerous). Similarly the word خوش نصیب (KhushNasib, Lucky). The prefixes can be articles,

[4] http://www.cle.org.pk/software/langproc/urdunormalization.htm.

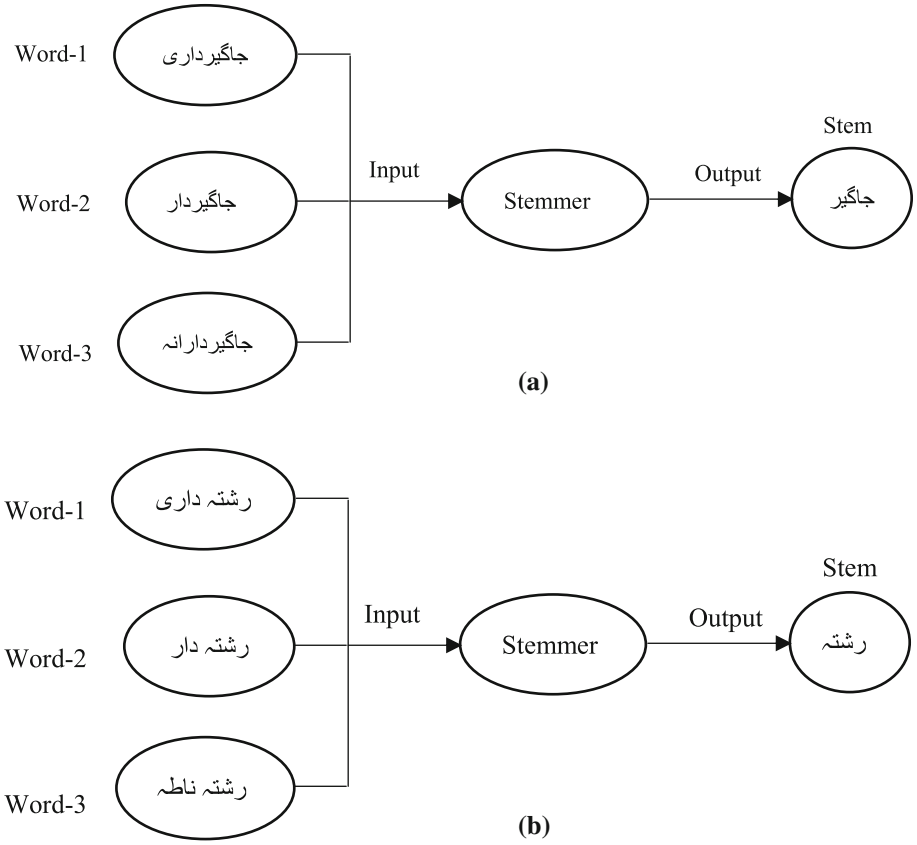

Fig. 3 **a** Stemming example of words having stem جاگیر (Jagir, Land). **b** Stemming example of words having stem رشتہ (Rishta, Proposal)

prepositions or conjunctions, whereas the suffixes are generally objects or personal / possessive anaphora. Both prefixes and suffixes are allowed to be combinations, and thus a word can have zero or more affixes.

The process of stemming improves IR systems query and documents matching ability. For example, we have Urdu words shown in Fig. 3. (a) جاگیردارانہ (Jageer darana, Feudal) and جاگیردار (Jageer dar, Landlord) after apply stemming algorithm the word is reduced to the root word جاگیر (Jageer, Land) similarly the words shown in Fig. 3. (b) رشتہ دار (Rishta dar, Relative) and رشتہ داری (Rishta dari, Relationship) رشتہ ناطہ (Rishta Nata, Kinship) the root word of these three words is رشتہ (Rishta, Relationship). From computational context, stemming process helps in enhancing recall because somebody searching for the words جاگیردارانہ (Jageer darana, Feudal) and رشتہ دار (Rishta dar, Relative) is most probably looking for words جاگیر (Jaagir, Land) and رشتہ (Rishta, Relationship) also.

Stemming is similar to the morphological analysis in NLP, but is used to attain somewhat different goals (Estahbanati and Javidan 2011). There is a little need of stemming in languages that has little inflectional element such as Mandarin Chinese (Riaz 2007). Stemming is also known as conflation. Stemming can be divided into two types (1) strong stemming and (2) weak stemming. Stemmer rules vary from language to language.

Initially, challenges in writing a stemmer for Urdu language are investigated by Riaz (2007). He said Urdu stemming is challenging due to (1) diverse nature of Urdu (2) the fact that Farsi and Arabic stemmers cannot be used for Urdu and (3) dictionary based correction methods cannot be used for Urdu due to lack of machine readable resources. A prototype based on four rules to process plurals and possessives with a heuristic to skip words which do not need stemming was proposed. It is observed that the order in which rules are executed is important as improved results are obtained by changing order. As the focus of the work was to investigate the challenges regarding Urdu stemmer so prototype is not optimized, however evaluation of prototype for IR task by using precision and recall metrics was considered as future work.

Akram et al. (2009) was unaware of the initial stemming work done by Riaz (2007). They stated that no work is reported on Urdu stemming and proposed a rule-based stemmer named Assas-Band. An enhancement in the performance of stemmer is achieved using precise affix-based exception lists as compared to conventional lexical lookup used for developing stemmers in other languages.

Khan et al. (2012) proposed a light weight stemmer for ULP. It has the capability to handle inflectional morphology and stem of a word was attained by removing prefixes and suffixes from a word. The stemmer achieves 73.55 % precision, 90.53 % recall and 81.16 % F1-Measure.They tested their proposed lightweight Urdu stemmer on their own constructed three different corpora i.e. corpus-1, corpus-2 and corpus-3 of size 9200, 27,000 and 30,000 words, respectively. Data in these corpora are organized in the form of verbs, nouns, adjectives, punctuations, numbers, special symbols etc. Gupta et al. (2013) developed a rule base Urdu stemmer and evaluated its performance on IR task. They tested their proposed Urdu stemmer on 2000 words. The affix list used in their experiment consisted of 119 rules. Their proposed system achieved up to 86.5 % accuracy.

Recently, Ali et al. (2014) proposed a novel rule base stemmer for Urdu. It uses affix stripping technique to generate stem of word. Their rules were mainly consisted of two types: prefix and postfix rules. They tested their stemmer on four datasets. Its performance is compared with Light Weight stemmer (Khan et al. 2012) and better results are shown. Actually the corpus used in their experiment is also constructed by them. The size of the four corpora used in Ali et al. (2014) experiments is: corpus-1(15,200 word), corpus-2(7250 words), corpus-3(24,238 words) and corpus-4(32,388 words).Corpus-1 contains data from politics and weather news domain, corpus-2 contains data from sports and terrorist related news domains, data of corpus-3 is collected from various grammar books and Urdu dictionaries while corpus-4 contains comprehensive news headline data from corpus-1, corpus-2 and corpus-3.

5 Techniques

The techniques developed for handling tasks in ULP can be categorized into three types (1) Rule-based (2) Statistical and (3) Hybrid.

Performance of any NLP techniques whether rule-based, statistical or hybrid, are usually evaluated by using Precision, Recall, Accuracy and F-measure. Recall and precision are inversely related as recall increases precision decreases. The F-measure is defined as a harmonic mean of precision (P) and recall (R). In context of IR the formulas of Precision, Recall, Accuracy and F-measure are given below.

$$Precision\,(P) = \frac{tp}{tp + fp} \tag{1}$$

$$Recall\,(R) = \frac{tp}{tp + fn} \tag{2}$$

$$Accuracy\,(A) = \frac{tp + tn}{tp + tn + fp + fn} \tag{3}$$

where *"tp", "fp", "fn" and "tn" stand for* True Positive, False Positive, False Negative and True Negative, respectively.

$$F-Measure\,(F) = \frac{2PR}{P + R} \tag{4}$$

5.1 Rule-based techniques

Rule-based techniques are based on set of rules or patterns which are defined to perform various NLP tasks. Akram et al. (2009) developed an Urdu stemmer named Assas-Band using rules. It maintains an Affix Exception List and works according to the algorithm to remove inflections. First the prefix is removed from the word which returns stem-postfix sequence. Then the postfix is removed and stem is extracted. Its reported accuracy is 91.2 %. Riaz (2010) presented a rule-based NER algorithm for Urdu. For experiments he chose 2262 out of 7000 document from Becker–Riaz dataset and refined various XML tags and its contents for readability. He experimentally showed that his rule-based technique for NER outperforms the models that use statistical learning. The results show the recall of 90.7 % and precision of 91.5 %.These results are still the best results reported on Becker–Riaz dataset using rule-based technique. Lehal et al. (2012) presented a rule-based Urdu Stemmer in which affixes removal are carried out from inflected words with the help of manually synthesized rules. They tested it on various Urdu news documents containing 20,583 words and reported 85.14 % accuracy. The problem with rule-based techniques is that they lack the robustness and portability (Chiong and Wei 2006). When new rules need to be introduced for some new information or new domains, rule-based techniques incurs sharp maintenance cost. Second problem with rule-based techniques is that that they are domain specific and one should have knowledge about the language as well as grammar rules.

5.2 Statistical techniques

The current dominant technique in NLP is supervised statistical learning. Most of the statistical learning models have the capability to automatically induce rules from training data. Statistical techniques are essential tools for analyzing large datasets. The technology for statistical NLP basically evolved from Machine Learning (ML) and DM. Statistical techniques use parametric, non-parametric or kernel based learning algorithms. In general, the parameters of a statistical model are trained on a dataset, and then the models are applied to different datasets for various NLP tasks and performance is observed. One important fact about statistical models is that, many statistical models are very much dependent on the training corpus. Different statistical techniques e.g. Conditional Random Fields (CRF), (SVM), N-Gram TnT taggers (Trigrams'n'Tags) and many others have been adopted to address the major NLP tasks, such as, NER, Word Segmentation, POS Tagging, Sentence Boundary Detection (SBD) and Parsing etc. Ekbal et al. (2008a) have developed a statistical Conditional Random Field (CRF) model for the development of NER system for South and South East Asian languages, mainly for Bengali, Hindi, Telugu, Oriya and Urdu. The rules

for identifying nested NEs for all the five languages and the gazetteer lists for Bengali and Hindi languages were used. The reported system achieved F-measure of 59.39 % for Bengali, 33.12 % for Hindi, 28.71 % for Oriya, 47.49 % for Telugu and 35.52 % for Urdu. Ekbal et al. (2008b) have used the IJCNLP2008 dataset for their experiments. The results of Ekbal et al. (2008a) are considered the lowest results reported using statistical technique for Urdu NER. Mukund and Srihari (2009) used a conditional random field (CRF) based technique for Urdu NE tagging. It is shown that by increasing the training data for POS learning by applying bootstrapping techniques improves NE tagging results. They show F-Score of 68.9 %. The datasets used in their experiments are CRULP and CRL. These results are considered ever first good results reported through statictical technique for Urdu NER task. Sajjad and Schmid (2009) have provided comparison of four statistical taggers namely TnT tagger, TreeTagger, RF tagger and SVM. They used tagged data of 107,514 words in their experiments for tagger comparison. For training they have taken 100,000 words and remaining data is used as test data. They experimentally have shown that that SVM tagger is the most accurate, showing 94.15 % correct prediction of tags. Remaining three taggers have accuracies of 93.02 % (Tree tagger), 93.28 % (RF tagger) and 93.40 % (TnT tagger).

The viability of statistical techniques depends on the existence of large size corpora for training phase. The more the training data the more promising results and vice versa. Also key to the successful use of statistical techniques especially CRF and MaxEnt is the design of an appropriate feature set. As in statistical learning most of the intelligence lies is in the feature extraction. So there is an ultimate need to craft a good feature vector that is highly relevant to the task.

5.3 Hybrid techniques

Hybrid techniques usually share features of both rule-based as well as statistical techniques. Like rule-based techniques they use predefined hand crafted rules for various NLP tasks and like statistical techniques they use ML models which automatically induce rules from training data. Chiong and Wei (2006) experimentally showed that the results produced by their proposed hybrid technique for NER which consists of Maximum Entropy (ME) and HMM are better than that of using single statistical model. Hybrid techniques usually outperform both rule-based and statistical techniques in the context of controlling sparseness in the data to some extent. Most of the recent NLP research works use hybridization to achieve better results (Rehman et al. 2011). Rehman et al. (2012) presented a hybrid technique for Urdu sentence boundary disambiguation consists of unigram statistical model and rule-based algorithm. They obtained 99.48 % precision, 86.35 % recall and 92.45 % F1-Measure. Lehal (2013) presented a hybrid technique for segmentation task, which uses top down mechanism for line segmentation and bottom up design for segmenting the line into ligatures. They classify the components correctly with 99.02 % accuracy.

Careful analysis of historical paradigm of tasks and techniques provides us with several interesting trends. One can see from Table 5 that not much work is done about ULP so far. ULP got attention of few researchers before 2010 and several rule-based techniques are used to perform different tasks. Especially statistical techniques are employed for ULP after 2010. The usage of statistical learning models needs special attention. Finally, hybridization of rule-based and statistical techniques can also provide state-of-the-art performances for different ULP tasks.

Table 5 Historical paradigm of techniques and tasks; rule-based (RB), statistical (St), hybrid (Hy), X means technique

Year/task	NER			POS tagging			Tokenization			SBD			WordNet			Stemming			Stop words		
	RB	St	Hy	RB	St	Hy	RB	St	Hy	RB	St	Hy	RB	St	Hy	RB	St	Hy	RB	St	Hy
2002	X						X														
2003				X																	
2007					X											X			X		
2008		X																			
2009		X					X			X						X					
2010	X	X					X														
2011							X					X	X								
2012		X			X								X			X					
2013									X							X					
2014					X											X					
2015											X										

6 Tasks

Following is a list of some of the most commonly investigated tasks in ULP.

6.1 Sentence boundary detection (SBD)

SBD is a preliminary step for preparing a text document for NLP tasks, e.g., machine translation, POS tagging, text summarization, IR, Parsing, chunking and so forth (Rehman et al. 2011; Wong et al. 2014). Sentence is a collection of words that gives a complete thought, and consists of subject and predicate, normally subject consists of one or two words, usually noun or a pronoun, while predicate indicates the action. Finding of sentence boundary in English is relatively easy as compared to Urdu.

Capitalization is the most important feature, which plays a very vital role in identifying sentence boundary in English. Unfortunately, Urdu and other Arabic script languages do not have this distinctive feature (Jawaid and Ahmed 2009).

SBD is a difficult task as very often ambiguous punctuation marks are used in the text. For example, generally a period "." in English appears at the end of the sentence as well as decimal in numbers, in email addresses, abbreviations and many more. Likewise question mark "?" and sign of exclamation "!" also appear inside the sentence. Ellipses and quotations are also used in the text and they too add ambiguity to sentence terminator marks. Similarly the punctuation mark dash '–' is used to identify the range of values, in dates and part of abbreviation. Examples mentioned in Tables 6, 7, 8, 9, 10 highlights the use of '–' in Urdu text. اسلم پچھلے تین –چار سالوں سےبے روزگاری کی وجہ سے ملک سے باہر ہے۔

۱۴-۰۸-۱۹۴۷ کو پاکستان آزاد ہوا۔

لوڈ شیڈنگ میں یو.پی.ایس ایک نعمت ہے۔

Although, in Urdu the character ع (Ain) is reserved to represent decimal point in numeric value but using dot "." an English character, to represent decimal point in Urdu numeric values is also a common practice. E.g. ۷ع۵ and ۷٫۵ are same terms. ریکیٹر سکیل پر زلزلے کی شدت ۷٫۵ تھی۔

The punctuations are also used in the middle of the sentence for different purposes (Jawaid and Ahmed 2009). احمد نے بتایا، "پاکستان میچ جیت گیا ہے"۔

Table 6 Example of '–' to identify the range of values

ہے	باہر	سے	ملک	سے	وجہ	کی	بےروزگاری	سے	سالوں	چار	-	تین	پچھلے	اسلم
hay	Bahir	sy	mulk	say	Waja	ki	Berozgari	say	salun	char	-	teen	pechlay	Aslam
Aslam is outside his country for the last three to four years due to unemployment.														

Table 7 Example of '–' used in dates

ہوا	آزاد	پاکستان	کو	۱۴	-	۰۸	-	۱۹۴۷
Howa	Azad	Pakistan	Ko	14	-	08	-	1947
Pakistan became independent on August 14, 1947.								

Table 8 Example of '–' used in parts of abbreviations

ہے	نعمت	ایک	ایس	-	پی	-	یو	میں	لوڈشیڈنگ
hay	naimat	aik	ais	-	pee	-	you	mein	Load shedding
UPS is blessing during Load shedding hours.									

Table 9 Example of '.' used in the numeric value in Urdu

تھی	۷.۵	شدت	کی	زلزلے	پر	سکیل	ریکٹر
thi	7.5	shidat	Ki	zalzalay	per	scale	Rector
The earthquake had a magnitude of 7.5 on the Richter scale.							

Table 10 Example of punctuation

“	ہے	گیا	جیت	میچ	پاکستان	"	،	بتایا	نے	احمد
“	hay	gaya	jeet	Match	Pakistan	“	,	bataya	nay	Ahmad
Ahmed told Pakistan have won the match.										

Although, SBD in ULP have significance importance for text processing task. Unfortunately, no much research has been conducted to address this task. Initially, Jawaid and Ahmed (2009) have only provided a detailed analysis of the main issues rather than its solutions. Consequently, the only pioneering work that addresses the subject area well is Rehman et al. (2011). They presented a hybrid technique for Urdu SBD having combination of unigram statistical model and rule-based algorithm. The authors first train the unigram model on tagged data, then the trained model is used to identify word boundaries in test data. Since the result generated by unigram model has low precision, so to overcome the problem of low precision the authors uses an algorithm based on handcrafted rules. A very basic unigram statistical model is used, while complex statistical models still need to be explored.

More ambiguities exist in Urdu because of the absence of space and case discrimination. Case discrimination and smooth use of space between words are powerful clues to identify sentence boundary in many languages. For example, in English a period followed by a space character and a word starting with an upper case letter, is a strong candidate to be a sentence marker. Urdu follows the unicameral script of Arabic, with or without space between words. Sometimes the use of space depends on the nature of the character a word ends with (joiner or non-joiner), space is used only after the words ending with a joiner character (Raj et al. 2015). Recently, Raj et al. (2015) presented Feed Forward Neural Network based Urdu SBM technique. It produced 93.05 % precision, 99.53 % recall and 96.18 % f-measure with varying size of data and threshold values. They have not used cross folding verification and also mixed the results of training and testing data.

6.2 Tokenization or word segmentation

The first step in the IR task is word segmentation or tokenization. Word segmentation is the foremost obligatory task in all NLP applications. The initial phase of text analysis for any language processing task usually involves tokenization of the input into words (Becker and Riaz 2002; Al-Shammari 2008; Durrani and Hussain 2010). Wrong tokenization produces wrong results. This task is non-trivial for the scarce resource languages such as Urdu, as there is inconsistent use of space between words. In English spaces are used to indicate word boundaries and this makes tokenization task easy, but in some case there is exception, e.g. tokenizing the word "can't" into its component words "can" and "not". In Chinese and Japanese, there are no spaces between words, and in Korean and Thai, spaces define words inconsistently. Urdu is morphologically rich language with different nature of its characters. Urdu text tokenization and sentence boundary disambiguation is difficult as compared to languages like English (Rehman et al. 2011). In languages like English, French, Hindi, Napali, Bengali, Greek, Russian etc. Space, Comma and semi colon are used for identifying a word boundary. But in Asian Languages, like Urdu and Chinese Space is not used consistently. Hence in some

Table 11 List of joiners and non-joiners alphabets

Isolated shapes of all Non-joiner Alphabets	ا،د،ڈ،ذ،ر،ڑ،ز،ژ،و،ے
Isolated shapes of all Joiners Alphabets	ب،پ،ت،ٹ،ث،ج، چ، ح، خ،س،ش،ص،ض،ط،ظ،ع، غ،ف،ق،ک،گ،ل،م،ن،ہ،ی،ھ

Table 12 Joiner and non-joiner example

Ghain(غ)Joiner	Example	Dal(د)Non-Joiner	Example
Initial shape	غیرت (Ghairat, Honour)	Isolated shape	د یکھنا (Deikhna, Watching)
Middle shape	بغیر وردی (BaaghairWardi, Without Uniform)	Final shape	بندر (Bundar, Monkey)
Final shape	بالغ (Baligh, Adult)	----	
Isolated shape	ابلاغ عامہ (Iblagh, Media)	----	

cases it can't be used for delimiting, making the segmentation challenging. Chinese text is written and printed without any space between words, which can be used in alphabetically written languages to identify word boundaries. One has to use high level information such as: information of morphology, syntax, statistical analysis and semantics of the language for word segmentation (Lehal 2010). There are several issues in tokenization in Urdu of which space insertion and space exclusion are important ones. Riaz (2010) address the challenges of Urdu NER and differentiate it from other South Asian (Indic) languages. He mentioned that among the other challenges for Urdu NER there is also a space exclusion challenge. To handle the challenge he proposed that in such a case rules are modified to recognize both occurrences separated with space. There are numerous tokenization techniques available for the various languages of the world, e.g., feature based techniques (Meknavin et al. 1997), rule-based techniques (Zhou and Liu 2002; Kaplan 2005) and statistical techniques (Yang and Li 2005). Significant work has also been done for Arabic (Attia 2007) which is closer to Urdu because of the same script.

6.2.1 Space insertion

There are two types of characters in Urdu, i.e. joiner and non-joiner characters (Rehman et al. 2011). Joiner characters are those characters which join with the next character. Joiners can acquire four different shapes namely initial, medial, final and isolated. The alphabet Ghain (غ) can take initial: غ , medial:غ, final: غ and isolated: غ.Similarly the alphabet Hay (ہ) take initial form: ھ , medial form: ھ , final form: ہ and isolated form: ہ . Non-joiner characters are those characters which do not join with the next character. Table 11 shows list of Joiners and Non-joiners Alphabets.

The non-joiner characters have the specialty that they can acquire only the final and isolated shapes. Arabic Letter Dal can only take final: د and isolated د. Non-joiners cannot acquire the initial or middle shape. These shapes are called glyphs. Glyph is a shape that a character can take in the text, e.g. initial, middle, final (Table 12).

As there is no concept of space in hand written Urdu text (Jawaid and Ahmed 2009; Lehal 2010; Durrani and Hussain 2010), e.g. آبی پرندے (Abi Parinday, water birds) for native speaker of language this is a single word, but in computer application these are two words and must be separated by the space. These challenges have been highlighted in Urdu text tokenization (Jawaid and Ahmed 2009). So far in Urdu language, word segmentation, faces space omission as well as space insertion error challenges. To address these two core problems Durrani and Hussain (2010) initially, discussed how orthographic and linguistic features

in Urdu trigger these two problems and then presented a hybrid algorithm as a solution. Durrani and Hussain (2010) segmentation model for space omission was based on maximum matching technique. They ranked the resulting probabilities by using min-word, unigram and bigram techniques. Space insertion problem was handled by using linguistic information to sub-classify the problem and then used different techniques for different cases such as affixation, reduplication, abbreviation and compound word. The overall accuracy reported by their segmentation model is 95.8 %.

6.2.2 Space exclusion

Space exclusion is another issue in Urdu text tokenization. Sometimes the space that separate the words, comes within the words and that group of words give a collective meaning representing one thing i.e. آبی پرندے (Abi Parinday, Water birds). If the space is not given in the word then it would look like this آبیپرندے (AbiParnday, Water birds), which will be not even understandable to the native speaker of the language (Jawaid and Ahmed 2009).

For space exclusion or omission issue in Urdu word segmentation Durrani and Hussain (2010) have used rule-based maximum matching technique to generate all the possible segmentations. Lehal (2010) have presented a very unique and interesting technique for space exclusion. They used Hindi for segmenting Urdu text after transliteration, because Hindi uses spaces consistently as compared to Urdu which has both space exclusion and insertion problems.

In the following cases the space should be neglected.

- Compound words
- Reduplication
- Affixation
- Proper Nouns
- English words
- Abbreviations and acronyms

Compound words

In Urdu followings are the categories of compound words (Jawaid and Ahmed 2009).

- AB
- A-o-B
- A-e-B

In AB form of words like, محنت مشقت (Mihnat Mushaqat, Hard Working), ماں باپ (Maan Baap, Parents) should be takes as one string. In A-o-B formation words like, عزت و حرمت (Izza to hurmat, Honour and Dignity) this should be considered as a single token. Third form of compound words is A-e-B i.e. طالب علم (Talib-e-Ilam, student) The combing mark[5] "zair" under Talib will make it a compound word. Without the "zair" they are two separate words.

Reduplication

Reduplication should also be considered semantically as a single unit. Reduplicates words are صبح صبح (Subh Subh, Morning), دن بدن (Din ba Din, Day by Day) are the reduplication words and are separated by the space (Jawaid and Ahmed 2009).

Affixation

Affixations are used in Urdu, both as prefixes and suffixes. Words like انتھک (Anthak, Tireless), خوش اخلاق (KhushI khlaq, Affable) are the examples of prefixes and should be treated

[5] The diacritics (called zer-e-izafat or hamza-e-izafat) are optional, and are not written in the example given.

as a single unit. The words like سرمایہ کاری (Sarmaya Kari, Investment), بد گمانی (Bud Gumani, Suspicion) are the examples of suffixes and should also be treated as a single unit (Akram et al. 2009; Estahbanati and Javidan 2011).

Proper Nouns

Proper nouns often consist of two or more parts, i.e. first name and last name (Jawaid and Ahmed 2009) and are often separated by the space but represent only one entity, e.g. اسلام آباد (Islamabad), سعودی عرب (Saudi Arab, Saudi Arabia), and حسن علی (Hassan Ali).

English Words

Some English language words are used in Urdu which includes spaces in between them (Jawaid and Ahmed 2009). Examples of these words are نیٹ ورک (NetWork, Network) and فٹ بال (Foot Ball) space must be ignored to consider them as a single entity.

Abbreviations and Acronyms

English abbreviations are used in Urdu, in the form of pronunciation of English character written in Urdu (Jawaid and Ahmed 2009). Examples of abbreviations are پی ایچ ڈی (PhD), and این ایل سی (NLC).

6.3 Part of speech (POS) tagging

POS tagging is done by taggers which are set of rules and their task is to, for given text, provide for each word its contextually disambiguated POS tag representing the word's morpho-syntactic category (Horváth 1999).

In order to increase the robustness and accuracy of any NLP System we need to depend upon its ability that how efficiently and accurately it extracts related data from a training corpus. The more accurate and related data an NLP system can extract the more vigorous and precise it is.

The accuracy of a statistical model developed for POS tagging not only depends on the domain of the dataset used to train the model but also on the tagset used for annotation (Mukund and Srihari 2012). Essentially, two things are needed to construct a POS tagger: a lexicon that contains tags for words and a mechanism to assign tags to running words in a text (Biemann 2006). POS tagging plays an important role in various applications like speech recognition, information extraction, text-to-speech and machine translation systems (Anwar et al. 2007). POS task is more challenging in languages which have rich morphology. It is customary that in languages having rich morphology, many words have more than one POS tags, which makes it tagging a crucial process.

The most frequent errors with automatic tagging is to differentiate between noun and the other open class tags in the noun phrase like proper noun, adjective and adverb. In Urdu, It is hard to determine a noun from proper noun, although there is a clear distinction between noun and proper noun. Acquiring distinctive contextual information for Urdu language is a difficult task (Naz et al. 2012). Table 13 provides corresponding POS tags from CLE[6] POS tagset for each word.

Like other languages the phenomenon of dropping of words is also frequent in Urdu. If a noun in a noun phrase is dropped, the adjective becomes a noun in that phrase.

The above table shows the occurrence of adjective with noun, and in the second phrase dropping the main noun from the noun phrase; in that case the adjective becomes a noun.

POS tagging task is more challenging in a language that have no or less annotated corpora (Graça 2011). There are three main tag sets designed for Urdu, the CRULP tagset, U1-tagset

[6] http://www.cle.org.pk/software/langproc/POStagset.htm.

Table 13 Example of multiple tags ambiguity

؟	کیا	کیا	کیلیے	آپ	نے	اس
?	Kiya	kaya	kayliyay	aap	nay	Us
PU	VBF	RB	NN	PRP	PSP	PRP
اس/PRP نے/PSP آپ/PRP کیلیے/NN کیا/RB کیا/VBF ؟/PU						
What has he done for you?						

and the tagset proposed in Sajjad (2007) referred as Sajjad tagset. The U1 tagset, released as a part of EMILLE corpus, is based on the EAGLES standards (Baker et al. 2003).

Several techniques have been applied for POS tagging. The first POS tagger for Urdu was developed by Hardie (2003). He discussed several problems of Urdu and developed a tagset for Urdu using the EAGLES guidelines for morpho-syntactic annotation of dataset. The guidelines were actually written for European Union languages but were easily applicable to Urdu due to Urdu's structural similarities with Indo-European family. This tagset can be considered as a beginning for the creation of necessary resources for Urdu POS tagging. It uses grammar of Urdu by Schmidt with the EAGLE guideline morpho-syntactic annotation. It has uni-rule disambiguator having approximately 270 written rules. It has an accuracy of about 90 % with a very high ambiguity level and 2.5 tags per word. The main drawbacks of rule-based systems are the laborious work of manually coding the rules and the requirement of linguistic background.

A statistical technique named n-gram Markov model is trained for tagger development and high accuracy is achieved when tested on two types of tagset (Anwar et al. 2007). Naz et al. (2012) developed a tagger using the statistical technique for the Urdu Language. They used Brill's transformation based learning, which deduces rules automatically from the training corpus and the accuracy achieved by employing this technique was comparable to the other statistical techniques.

Recently, Abbas (2014) presented a semi-semantic POS annotation and its evaluation via Krippendorff's 'α' for Urdu. KON-TB treebank developed for Urdu. To achieve high annotation quality dataset was annotated manually. The size of the dataset used in their experiment was limited to 1400 sentences and after evaluation their inter-annotator agreement obtained is 0.964 %.

6.4 Named entity recognition (NER)

The core objectives of the most information extraction applications are the detection and classification of the named entities in a text. Named entity means anything that can be referred to with proper name. The process of NER refer to the combined task of finding extents of text that constitute proper names and then classifying the entities being referred to according to their types.

The NER task came into focus during the sixth Message Understanding Conference (MUC-6). After that many NER systems were developed. Most of these systems were developed for European languages. For south Asian languages, NER systems are yet in developing phase. IJCNLP-08 workshop played a major role in development of NER Systems for Indian languages including Urdu. This Workshop focused on five languages i.e. Hindi, Bengali, Oriya, Telugu and Urdu. All the systems were developed using Statistical techniques or Hybrid techniques (Singh et al. 2012).

At the sixth conference (MUC-6) the task of NER was defined as three subtasks: ENAMEX (for the person, location, and organization names), TIMEX (for date and time expressions), and NUMEX (for monetary amounts and percentages). Tables 14 and 15 provide lists of

Table 14 Example of syntactic ambiguity

دو	خوراک	کو	لوگوں	ضرورتمند
Do	khorak	ko	logoon	Zaroratmand
VB	NN	P	NN	Adj
Give	Food	To	People	Needy
Give Food to needy people				
	دو	خوراک	کو	ضرورتمند
	do	khorak	ko	Zaroratmand
	VB	NN	P	NN
	Give	Food	To	Needy
Give Food to needy				

Table 15 List of generic Urdu named entity types with the kinds of entities they refer

Type	Tag	Sample categories
People	PER	Individuals, small groups
Organization	ORG	Companies, political parties, agencies, religious groups, sports team
Location	LOC	Countries, states, provinces, physical extents,mountains,lakes, seas, bridges, buildings, airports
Date	DATE	Year and months
Time	TIME	Seconds, minute, hours and periods of time
Designation	DESIG	Various designations e.g. Prime Minister, President, Chief Justice, Captain
Number	NUM	One, two, one thousand, one million

typical NE types and their examples as mentioned in Riaz (2010), Singh et al. (2012) (Table 16).

Conditional Random Fields (CRF) was employed to develop NER system for South and South East Asian languages especially Hindi, Urdu, Bengali, Telugu (Ekbal et al. 2008b). Rules for identifying nested NEs for all the five languages were used. The reported system achieved F-measure of 35.52 % for Urdu, 59.39 % for Bengali, 28.71 % for Oriya and 4.74 % for Telgu. Later, Mukund and Srihari (2009) presented conditional random field (CRF) based technique for Urdu NE tagging. Their proposed four stage model shows F-Score of 68.9 % for NE tagging which is much higher as compared to the results reported in their early attempt using two stage model.

NER is a difficult task to be handled in languages that do not have large annotated corpora. Automated text processing needs NER as a vital part in NLP, intelligence gathering and Bioinformatics (Riaz 2010). Common entities in Bioinformatics domain include genes, protein, disease, drugs, body parts, etc. Text processing applications, such as MT, IE, IR or NLP require recognizing; names, numbers, organizations and geographical places (Riaz 2010). NER goal is to recognize these entities. Finding these entities plays key role in information management for specific applications. When the person or organization is more important than the action it performs, NER becomes inevitable (Riaz 2010).

NER is focused on finding the name entities in a text and then disambiguating them. Structural and semantic ambiguities are its two important types (Matsukawa et al. 1993). NER in English and European languages is researched a lot as compared to South Asian languages. Lack of POS taggers, gazetteers and majorly large annotated corpora are key hurdles. Becker and Riaz (2002) done an initial NER study for South Asian languages which includes Urdu, which also resulted in the creation of a popular Urdu Corpus used frequently. Generally, NER is a tough task in all languages but it's harder in Urdu which lacks the most vital Capitalization feature. English and European languages NER key feature to recognize

Table 16 Named entity types with examples

Type	Example
People <PER>	احمد شہزاد <PER>/</PER>بھی زیادہ دیر کریز پر نہ ٹھہر سکے اور <PER>جیمز اینڈرسن</PER> کی گیند پر کیچ ہوگئے۔ <PER>Ahmed Shehzad</PER> not stay too long at the crease and got off onthe bowling of <PER> James Anderson</PER>.
Organization <ORG>	ورلڈ اکنامک فورم <ORG>/</ORG> نے جنسی مساوات پر اپنی رپورٹ میں کہا ہے <ORG>World Economic Forum</ORG> said in its report on gender equality---
Location <LOC>	برطانیہ <LOC>/</LOC> کے سابقہ جیمز برابکاکہنا ہے <LOC>Britain's</LOC> former army chief says …
Date <DATE>	برطانیہ کے فوجی حکام کا خیال تھا کہ <DATE>/</DATE> سنہ دو ہزار پانچ تک عراق کی صورتحال میں بہتری آئے گی UK military officials believed that up to<DATE> 2005</DATE> Iraq would improve the situation.
Time <TIME>	پاکستان اور انڈیا کے درمیان میچ کل دن <TIME>/</TIME> نو بجے شروع ہوگا Match between Pakistan and India will begin tomorrow at <TIME>nine o'clock</TIME>
Designation <DESIG>	پاکستان کے <DESIG>/</DESIG> وزیر اعظم نواز شریف آج سے تین روزہ دورے پر امریکا روانہ ہوئے Pakistan<DESIG> Prime Minister</DESIG> Nawaz Sharif left the United States on a three-day visit
Number <NUM>	محمد حفیظ کی <NUMR>/</NUM> ایک سو اٹھہتر <NUM> رنز ناٹ آوٹ کی شاندار اننگز نے ٹیم کی جیت میں اہم کردار ادا کیا۔ Mohammad Hafeez glorious unbeaten <NUM>178</NUM> -run innings played an important role in the team victory.

names is Capitalization, while orthography of Urdu does not support it e.g. in Urdu BBC is بی بی سی (Riaz 2010).

Ambiguity means one word or sentence representing more than one meaning i.e. Brown is a name and is a color similarly سحر (Sahar, Morning) in Urdu is name and also represents dawn.

A NER system can be rule-based, statistical or hybrid. A rule-based NER system uses hand-written rules to tag a corpus with named entities. A statistical NER system learns the probabilities of named entities using training dataset, whereas hybrid systems use both (Gali et al. 2008). Rule-based and statistical techniquess include many statistical models e.g. HMM, CRF, SVM, MaxEnt used to develop NER systems for Urdu but the results of rule-based techniques are more accurate as compared to statistical techniques in the context of NER task of Urdu (Riaz 2010; Singh et al. 2012).

6.5 Parsing

Parsing is the process of finding the integral structure of a sentence in a language by using grammar of the language. It is a paramount requisite for many language technology applications. It is one of the major tasks and core component in many systems for NLP, which helps in understanding the natural language (Jafar et al. 2004). The application areas where parsing plays a vital role include machine translation, word sense disambiguation, question answering, summarization and natural language text understanding. There is diversity of parsing techniques available, each of which suits particular situation. The Context Free Grammar (CFG) also known as rule-based grammar is the most common one; however statistical models based parsing techniques are also presented in literature.

The two most basic types of parsing are top-down and bottom up, however there are parsing algorithms that are of different type and there are some that are a mixture of these two. Top down parsers begin with the start symbol S (sentence) and stretch the sentence by applying the rules until the desired string is arrived. Bottom-up parsing as its name reflects works in opposite direction from top down parsing. Bottom up parsing starts building process from terminal leaf node and move in upward direction until the start symbol is reached. Along the way, a bottom-up parser searches for substrings of the working string that match the right side of some production. The parser reduces the substrings when some production rules and substring matching occurs. In short, the bottom-up parser begins with the strings (from the lowest part e.g. leaf node) and endeavors to fabricate a tree from the strings up (Mukhtar et al. 2012). The most commonly used technique for bottom up parsing is shift reduce parsing.

Kabir et al. (2002) proposed two phase parsing technique for sentence analysis in order to develop a Grammar Checker for Urdu. It has the capability to solve complex parsing problem in an efficient manner. The two main features of their proposed system are that, it provides facility to keep grammar simple; and gives you the facility of transformations in a simple way. It shows grammatical correction for declarative Urdu erroneous sentences.

Jafar et al. (2004) proposed a language specific parsing technique for Urdu sentences. Only morphologically closed classes of words, such as, conjunctions, postpositions, verb morphemes tags are inititated in it. Instead of simple CFG rules, lexical functional grammer was used for lexical and syntactic information. The method used in the development of parser for Urdu sentence was based on chunking, which utilizes linguistic characteristics of the morphologically closed classes in Urdu language. Single words with hard and soft spaces are hanlded by proposed tokenization algorithm.

Mukhtar et al. (2012) have proposed a new technique for developing a probabilistic Urdu parser, which was dependent on Probabilistic context free grammar (PCFG). It was based on multi-path shift reduce-strategy instead of breadth first strategy (BFS). Successive and pharases based rules were provided for a sentence to show its structure. Variables were used to hold rule's probalities. Rules probablities were added pathwise and highest probality parse tree was selected as accurate solution.

6.6 WordNet

WordNet is a lexical database or a large tree structured electronic dictionary for English (Miller 1995). About 155,000 nouns, verbs, adverbs and adjectives are present in it (Fellbaum 1998). Words are grouped into the set of synonyms called synsets or the words in the synsets are grouped according to the similarity in meaning e.g. رو (Row, Weep) and آنسوبہانا (AnsooBahana, Weep) has the semantically same meaning (Adeeba and Hussain 2011; Ahmed and Hautli 2011). WordNet is an important enabling technology for concept understanding and word sense disambiguation tasks.

For the development of sophisticated NLP techniques, it is required to have a rich lexical knowledge resource that can help by providing the meaning of a sentence through information on the lexical semantics of the words in a sentence (Ahmed and Hautli 2011). Existing research mainly focuses on English. Previously, no lexical knowledge base existed for Urdu (Adeeba and Hussain 2011; Ahmed and Hautli 2011). Consequently, they develop a first Urdu language WordNet. Hindi language processing can benefit from Hindi WordNet great source but it is not usable for Urdu (Riaz 2012). Most of the analysis of the words and the categorization of words in Hindi WordNet was done by using highbrow Hindi. For example, Urdu speakers are completely unfamiliar to Hindi WordNet terminology used to describe POS. Urdu uses Persian and Arabic based POS words as compared to Sanskrit based words used by Hindi (Riaz 2012). The noun word in Urdu is ism (اسم/) while in Hindi it is sangya and proper noun in Hindi is "Vyakti vachak sang" These differences are completely unknown by Urdu speaker who did not studied Hindi grammer. One needs to be expert of both languages, in order to deal with these differences (Riaz 2012).

Recently, Adeeba and Hussain (2011) and Ahmed and Hautli (2011) reported on development of a lexical resource for Urdu from Hindi WordNet; which currently contains about 50,000 unique words organized in 28,967 synsets; New words that do not exist in Hindi WordNet are also added, e.g, the word ربا (Reba, Usury) is not in Hindi WordNet. There is still need to add words from Persian and Arabic languages in the Urdu WordNet developed by Adeeba and Hussain (2011). Hindi WordNet was inspired by the

English WordNet although the script is different but both languages share the same structure. The scriptural barrier is crossed by using automatic and manual transliteration (Adeeba and Hussain 2011). It is available both for Urdu and Roman that makes it usable for non-Urdu speakers, who are not familiar with the Urdu script. Some words in Urdu are similar in writing but have completely different meanings. For example the word بنا (Ban-na) is written same for both ban-na (making) and bun-na (knitting) (Adeeba and Hussain 2011).

Building a WordNet complete in all aspects is a complex and difficult task. Zafar et al. (2012) proposed two techniques for building Urdu WordNet. Top-down technique, in which the source language is translated to target language. The synsets are translated from source to target by mapping the concepts. There limitation of this technique is that the target language must have synsets for each concept (Zafar et al. 2012). Second technique described by Zafar et al. (2012) is bottom up. This technique can either be used by merging or expanding (Thoongsup et al. 2009). The merge technique uses Princeton WordNet (PWN) and uses English dictionary to find the equivalent words from bilingual dictionaries. It works well for those languages which have similar scripts with English but creates more problems for those languages which have significantly different scripts e.g. South Asian languages. Another technique used is bottom-up expanding, which used bilingual dictionaries to directly map local words to PWN's synsets (Thoongsup et al. 2009).

7 Applications areas

In this section, we provide the impact of using ULP techniques for IR, classification and plagiarism detection areas. ULP techniques when properly applied do provide the basis for enhanced performance of different techniques.

7.1 Information retrieval (IR)

IR community is forty years old and numerous advances have been made. IR task is to retrieve documents related to input queries by the user. Major efforts have been made in English but right to left written languages such as Urdu has not got enough attention due to lack of resources.

Urdu IR was initially performed by using a small dataset that consists of 200 documents from the 7000 documents of Becker–Riaz dataset. The dataset was kept in Corpus Encoding Standard. The performance is measured using standard IR evaluation metrics precision and recall, without and with stemming and stop word removal, in which without stemming and stop word removal technique is considered as a baseline. Stemming and stop words removal are major preprocessing steps for indexing documents. Stop words are not useful for IR due to little sematic weight (Riaz 2008a). Stemming brings the word to root or stem of words which reduces index size so user doesn't need to worry about word variants while writing queries. Stemming helps in improving the IR performance especially in terms of recall (Pandey and Siddiqui 2009). Gupta et al. (2013) shown the effectiveness of rule-based Urdu stemmer for IR task. 119 rules are made for 2000 words dataset and 86.5 % accuracy is achieved. Becker–Riaz dataset is appropriate for IR techniques because it is comprehensive and consists of diverse topic news articles (Riaz 2008a).

7.2 Classification

Text classification is a major task in DM and receives considerable popularity for many applications in the real world e.g. document classification for electronic library shelf management. It is a process of classifying unknown text into an appropriate class to which it belongs.

Ali and Ijaz (2009) compared two major statistical techniques Support Vector Machines and Naïve Bayes for Urdu document classification. These classifiers are trained and tested by using a large corpus which was taken from Urdu news websites to achieve better accuracy. The dataset has been divided into six categories and has 19.3 million words. 90 % of the dataset was used as training set and remaining 10 % is used as test set. Standard preprocessing techniques like tokenization, stemming, stop words elimination, normalization and diacritics elimination were applied to get reduced feature lexicon. Experimental results using Precision, Recall and Accuracy show that Support Vector Machines performs better than Naïve Bayes. Additionally, it is seen that stemming is not useful in the classification but the removal of stop words increases the accuracy of the algorithms (Ali and Ijaz 2009).

7.3 Plagiarism detection

Plagiarism detection is one of the important application areas which use NLP techniques. It is about finding the copy pasted text in a document from other documents. Similar to IR and classification of English language was the main focus for plagiarism detection using NLP.

Recently, Khan et al. (2011) investigated the task of copy detection in Urdu documents and used n-gram model for word matching. They found that trigram model performs better for which the resemblance threshold was set to 75 %. N-gram model is applied on short text passages which indicate that result could be different if the trigram model is applied on long text passages. In the n-gram model only the punctuations are removed, no stemming is done while stemming plays important role to increase the accuracy of the algorithm. The performance of plagiarism detection techniques is usually evaluated using Precision, Recall and Accuracy.

8 Research issues and future directions

In this section, we will discuss research issues, open challenges and future directions in the field of ULP and its application areas. We categorize the challenges into four types.

The first type of challenges comes from the need of gold standard dataset preparation for all the problems of ULP. In some cases ULP is performed using statistical methods and rule-based techniques are found better, while this is not the case with other languages such as English. Less availability of large annotated datasets restricts ULP researchers to investigate the usefulness of statistical techniques for Urdu. The gold standard dataset provides a road map to do research about different tasks in a language as they are used to compare techniques in a fair manner. Less interest of researchers for ULP is one of the main reasons of not having large datasets for Urdu. There is a dire need of large annotated datasets of Urdu to apply statistical methods on different ULP tasks and see how they compare to already applied rule-based techniques. As rule-based techniques can perform well on small datasets but statistical methods need large annotated data to train in order to perform well.

The second type of challenges comes from the need of improved preprocessing for Urdu. Especially stemming needs special attention in terms of applying statistical methods for

improved Urdu stemmer. As previously only rule-based methods were explored and statistical learning was proved effective for making more accurate stemmers for other languages.

The third type of challenges come from the need of scalable and reliable rule-based, Statistical and hybrid techniques that resolve problems more efficiently in various ULP tasks e.g. POS, NER, SBD, Parsing and WordNet development etc. Some of the most popular statistical models such as Hidden Markov Model (HMM), Genetic Algorithm (GA), Neural Network, Decision Trees and Conditional Random Field (CRF), independently or jointly or have been implemented to process challenges effectively on NLP tasks in English, Arabic and French etc. However, some efforts are made for ULP on small datasets where rule-based techniques outperformed statistical techniques (Saeeda et al. 2014). Rule-based techniques usually perform better on small datasets but their performance degrades on large datasets due to absence of rules and scalability issue. Therefore in future statistical techniques need more research attention from ULP research community on large datasets.

The fourth type of challenges come from the need of improved NLP for application areas such as IR, classification, clustering, document summarization and plagiarism detection. In the past, they did benefit from ULP in most cases when rule-based techniques were used. There is still need to develop and investigate statistical ULP techniques for these application areas. IR methods need to be explored for Urdu such as N-Gram and WordNet which can consider the phrase structure and capture semantics, respectively. Naïve Bayes and Support Vector Machine models are used for classification Urdu text (Ali and Ijaz 2009). Both these models do not exploit the word dependencies and word semantics. HMM can be used to capture dependencies of words on previous words and CRF can be used to capture random dependencies between words for better classification. State-of-the-art topic model Latent Dirichlet Allocation can be employed to capture word semantics to overcome the problem of exact word matching problem (Daud et al. 2010). Plagiarism detection is performed using only stop word removal while stemming is ignored by Khan et al. (2011). Stemming, POS tagging, tokenization and consideration of semantics proved very useful for improved plagiarism detection in English language and are needed to be explored for Urdu language as well. Clustering is another major DM functionality which group text by using distance metrics in an unsupervised way. Due the unique nature of Urdu language it will be interesting to employ different NLP techniques and study their effect to cluster text and provide better Urdu text clustering methods with improved ULP techniques.

9 Conclusions

This paper introduces Urdu language and the complexities regarding its processing. The comparison between Urdu and other languages was discussed. Standard Text as well as lexical resources are paramount for carrying various NLP tasks in any languages of the globe. This work uncover the available Urdu linguistic resources e.g. the datasets and WordNet, which will help future Urdu researchers in conducting research and building other standard resources. Urdu orthography and morphology are described with the help of suitable examples. Moreover the solid discussion about Urdu characteristic and its resource sharing with Hindi emphasize on separate research for both. A review of techniques, for stemming, and taxonomy of different linguistic analysis, tokenization, SBD, POS tagging, NER, Parsing and WordNet for considering semantics is provided. Different application areas such as IR, classification and plagiarism detection also benefit from ULP. This paper provides the basis for developing latest techniques using statistical learning for ULP and emphasize on developing

large annotated datasets to compare the performance of rule-based and statistical methods. Statistical methods got a little attention for ULP due to less availability of large annotated datasets that are necessary for evaluating their performance. Collectively this paper provided an overview of research conducted about ULP, their impact on application areas and potential challenges. However, we do believe that the valuable information about ULP discussed here will be helpful for Urdu research community at present and in upcoming NLP era.

Acknowledgements The work is supported by Higher Education Commission (HEC), Islamabad, Pakistan.

References

Abbas Q (2014) Semi-semantic part of speech annotation and evaluation. In: Proceedings of ACL 8th Linguistic Annotation Workshop held in conjunction with COLING, Association of Computational Linguistics, pp 75–81

Adeeba F, Hussain S (2011) Experiences in building the UrduWordNet. In: Proceedings of the 9th workshop on Asian language resources, pp 31–35

Ahmed T, Hautli A (2010) Developing a basic lexical resource for Urdu using Hindi WordNet. In: Proceedings of CLT10, Islamabad, Pakistan

Ahmed T, Hautli A (2011) A first approach towards an UrduWordNet. Linguist Lit Rev 6(1):1–14

Akram Q, Naseer A, et al. (2009) Assas-band, an affix-exception-list based Urdu stemmer. In: Proceedings of the 7th workshop on Asian language resources, pp 40–46

Ali S, Khlid S, Saleemi MH (2014) A novel stemming approach for Urdu language. J Appl Environ Biol Sci 4(7S):436–443

Ali A, Ijaz M (2009) Urdu text classification. In: Proceedings of the 7th international conference on frontiers of information technology, pp 1–7

Al-Shammari (2008) Towards an error free stemming. In: Proceedings of ACM workshop on improving non English web searching, pp 9–16

Anwar W et al (2006) A survey of automatic Urdu language processing. In: Proceedings of conference on machine learning and cybernetics, pp 4489–4494

Anwar W, et al (2007) A statistical based part of speech tagger for Urdu language. In: Proceedings of IEEE international conference on machine learning and cybernetics, pp 3418–3424

Attia M (2007) Arabic tokenization system. In: Proceedings of the Urdu2007 workshop on computational approaches to semitic languages: common issues and resources, pp 65–72

Baker A, Hardie P et al (2003) Corpus data for south Asian language processing. In: Proceedings of the 10th annual workshop for South Asian language processing, pp 1–8

Becker D, Riaz K (2002) A study in Urdu corpus construction. In: Proceedings of Urdu 3rd workshop on Asian language resources and international standardization, pp 1–5

Biemann C (2006) Unsupervised part-of-speech tagging employing efficient graph clustering. In: Proceedings of the 21st international conference on computational linguistics and 44th annual meeting of the association for computational linguistics: student research workshop, pp 7–12

Capstick J, Diagne AK, Erbach G, Uszkoreit H, Leisenberg A, Leisenberg M (2000) A system for supporting cross-lingual information retrieval. Inf Process Manag 36(2):275–289

Chiong R, Wei W (2006) Named entity recognition using hybrid machine learning approach. In: Proceedings of international conference on cognitive informatics, pp 578–583

CLE (2015) Urdu digest POS tagged corpus. Retrieved 2015-08-07, from http://www.cle.org.pk/software/localization.htm

Daud A et al (2010) Knowledge discovery through directed probabilistic topic models a survey. Front Comput Sci 4(2):280–301

Durrani N, Hussain S (2010) Urdu word segmentation. In: Proceedings of international conference on human language technologies, pp 528–536

Ekbal A, et al. (2008) Named entity recognition in Bengali: a conditional random field approach. In: Proceedings of the 3rd international joint conference on natural language processing (ijcnlp), pp 589–594

Ekbal A, Haque R, Das A, Poka V, Bandyopadhyay S (2008). Language independent named entity recognition in Indian languages. In: Proceedings of the IJCNLP workshop on NER for South and SouthEast Asian languages, pp 33–40

Estahbanati S, Javidan R (2011) A new stemmer for Farsi language. In: Proceedings of international symposium on computer science and software engineering (CSSE), pp 25–29

Fellbaum C (1998). WordNet. Blackwell Publishing Ltd, New York
Flagship (2012) Undergraduate program and resource center for Hindi-Urdu at the university of Texas at Austin. Retrieved 2015-03-09, from http://HindiUrduflagship.org/about/two-languages-or-one/
Gali K, et al (2008) Aggregating machine learning and rule-based heuristics for named entity recognition. In: Proceedings of the ijcnlp-08 workshop on NER for South and SouthEast Asian languages, pp 25–32
Graça J et al (2011) Controlling complexity in part-of-speech induction. J Artif Intell Res 41(2):527–551
Gupta V, Joshi N, Mathur I (2013) Rule based stemmer in Urdu. In: Proceedings of IEEE 4th international conference on computer and communication technology (ICCCT), pp. 129–132
Gupta V, Joshi N, Mathur I (2015) Design and development of rule based inflectional and derivational Urdu stemmer 'Usal'. In: Proceedings of IEEE international conference on futuristic trends on computational analysis and knowledge management (ABLAZE), pp. 7–12
Hardie A (2003) Developing a tagset for automated part-of-speech tagging in Urdu. In: Proceedings of conference on corpus linguistics, Lancaster, pp 1–7
Henderson R, Deane S (2003) Xml made simple. Routledge
Horváth T et al (1999) Application of different learning methods to Hungarian part-of-speech tagging. Induc Logic Programm 1634(1):128–139
Humayoun M, et al. (2007) Urdu morphology, orthography and lexicon extraction. In: Second workshop on computational approaches to Arabic script-based languages,(caasl-2: Lsa), pp 1–8
Hussain S (2008) Resources for Urdu language processing. In: Proceedings of the 6th workshop on Asian language resources (IJCNLP'08), pp 99–100
Imran MR (2011) Online Urdu character recognition in unconstrained environment (doctoral dissertation, International Islamic University, Islamabad)
Jafar R, et al (2004) Language oriented parsing through morphologically closed word classes in Urdu. In: Proceedings of IEEE student conference on engineering, sciences and technology, pp. 19–24
Jawaid B, Ahmed T (2009) Hindi to Urdu conversion: beyond simple transliteration. In: Proceedings of the conference on language and technology, pp. 24–31
Kabir H, et al. (2002) Two pass parsing implementation for an Urdu grammar checker. In: Proceedings of IEEE international multi topic conference, pp. 1–8
Kaplan R (2005) A method for tokenizing text. CSLI Publications, Stanford, UK
Khan SA, Anwar W, Bajwa UI, Wang X (2012) A light weight stemmer for Urdu language: a scarce resourced language. In: 24th international conference on computational linguistics, pp 69–78
Khan M, et al. (2011) Copy detection in Urdu language documents using n-grams model. In: Proceedings of international conference on computer networks and information technology (ICCNIT), pp 263–266
Lehal, et al. (2012) Rule based Urdu stemmer. In: Proceeding of the 24th international conference on computational linguistics, pp 267–276
Lehal, G. (2010). A two stage word segmentation system for handling space insertion problem in Urdu script. In: Proceedings of the 1st workshop on south and southeast Asian natural language processing (WASSANLP), the 23rd international conference on computational linguistics(COLING), pp 43–50
Lehal, G. S. (2013). Ligature segmentation for Urdu OCR. In: Proceedings of IEEE 12th international conference on document analysis and recognition (ICDAR), pp. 1130–1134
Matsukawa T, et al. (1993) Example-based correction of word segmentation and part of speech labeling. In: Proceedings of the workshop on human language technology, pp 227–232
Meknavin S, et al. (1997) Feature-based Thai word segmentation. In: Proceedings of natural language processing Pacific Rimsymposium (NLRPS), pp. 35–46
Miller GA (1995) WordNet: a lexical database for English. Commun ACM 38(11):39–41
Mukhtar N et al (2012) Algorithm for developing Urdu probabilistic parser. Int J Electr Comput Sci IJECS-IJENS 12(3):57–66
Mukund, S., & Srihari., R. (2009). NE tagging for Urdu based on bootstrap POS learning. In: Proceedings of third international cross lingual information access workshop, pp. 61–69
Mukund S et al (2010) An information-extraction system for Urdu-a resource-poor language. ACM Trans Asian Lang Inf Process 9(4):1–43
Mukund S, Srihari R (2012) An NLP framework for non-topical text analysis in Urdu—a resource poor language (unpublished doctoral dissertation). State University of New York at Buffalo
Naz F et al (2012) Urdu part of speech tagging using transformation based error driven learning. World Appl Sci J 3(16):437–448
Naz S et al (2014) Challenges of Urdu named entity recognition: a scarce resource language. Res J Appl Sci Eng Technol 8(10):1272–1278
Paik J, et al. (2011). A novel corpus-based stemming algorithm using co-occurrence statistics. In: Proceedings of the 34th international ACMSIGIR conference on research and development ininformation retrieval, pp 863–872

Pandey AK, Siddiqui TJ (2009) Evaluating effect of stemming and stop-word removal on hindi text retrieval. In: Tiwary US, Siddiqui TJ, Radhakrishna M, Tiwari MD (eds) Proceedings of the first international conference on intelligent human computer interaction. Springer, pp 316–326

Prasad, K., & Virk., S. (2012). Computational evidence that Hindi and Urdu share a grammar but not the lexicon. In: Proceedings of the 24th international conference on computational linguistics (COLING), pp 1–13

Raj S, Rehman Z, Rauf S, Siddique R, Anwar W (2015) An artificial neural network approach for sentence boundary sisambiguation in Urdu language text. Int Arab J Inf Technol 12(4):395–400

Ranta A (2004) Grammatical framework: a type-theoretical grammar formalism. J Funct Programm 14(2):145–189

Rehman Z et al (2012) A hybrid approach for Urdu sentence boundary disambiguation. Int Arab J Inf Technol 9(3):250–255

Rehman Z, et al. (2011) Challenges in Urdu text tokenization and sentence boundary disambiguation. In: Proceedings of the 2nd workshop on South and Southeast Asian natural language processing (WASSANLP 2011), pp 40–45

Riaz K (2007) Challenges in Urdu stemming. In: Proceedings of BCS IRSG symposium on future directions in information access, pp 1–4

Riaz K (2008a) Baseline for UrduIR evaluation. In: Proceedings of the 2nd ACM workshop on improving on English web searching, pp 97–100

Riaz K (2008b) Concept search in Urdu. In: Proceedings of the 2nd PhD workshop on information and knowledge management, pp 33–40

Riaz K (2009) Urdu is not Hindi for information access. SIGIR workshop on information access in a multilingual World, pp 53–57

Riaz K (2010) Rule-based named entity recognition in Urdu. In: Proceedings of the 2010 named entities workshop, pp 12–35

Riaz K (2012) Comparison of Hindi and Urdu in computational context. Int J Comput Linguist Nat Lang Process 1(3):92–97

Rizvi, S., & Hussain, M. (2005). Analysis, design and implementation of Urdu morphological analyzer. In Proceedings of student conference on engineering sciences and technology (sconest), pp 1–7

Sajjad H (2007) Statistical part of speech tagger for Urdu. Master unpublished thesis: National University of Computer and Emerging Sciences. Lahore, Pakistan

Sajjad H, Schmid H (2009) Tagging Urdu text with part of speech: a tagger comparison. In: Proceedings of the 12th conference of the European chapter of the association for computational linguistics, pp 692–700

Sattar SA (2009) A technique for the design and implementation of an OCR for printed Nastaliq text. Doctoral dissertation, NED University of Engineering and Technology, Karachi

Schmidt R (1999) Urdu: an essential grammar (1st edn). British library catalog using in publication data: Routledge 11 New Fetter Lane, London EC4P 4EE

Singh U et al. (2012) Named entity recognition system for Urdu. In: Proceedings of international conference on Urdu, pp 2507–2518

Small and George (1908) A grammar of the Hindustani of Urdu language (30th edn). California digital library: London : K. Paul, Trench, Trübner Co., ltd

Thoongsup S et al (2009) Thai WordNet construction. In: Proceedings of the 7th workshop on Asian language resources, pp 139–144

Visweswariah K, et al. (2010) Urdu and Hindi: translation and sharing of linguistic resources. In: Proceedings of the 23rd international conference on computational linguistics (COLING), pp 1283–1291

Wong DF, Chao LS, Zeng X (2014) Isentenizer-μ: multilingual sentence boundary detection model. Sci World J 2014:1–10

Yang C, Li K (2005) A heuristic method based on a statistical approach for Chinese text segmentation. J Am Soc Inform Sci Technol 56(13):1438–1447

Zafar A, et al. (2012) Developing Urdu WordNet using the merge approach. In: Proceedings of conference on language and technology, pp 55–59

Zhang C, Baldwin T, Ho H, Kimelfeld B, Li Y (2013) Adaptive parser-centric text normalization. In: ACL (1), pp 1159–1168

Zhou L, Liu Q (2002) A character-net based Chinese text segmentation method. In: Proceedings of the Urdu 2002 workshop on building and using semantic networks, pp 1–6