IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 12, 2017, accepted December 20, 2017, date of publication December 28, 2017,
date of current version March 9, 2018.

Digital Object Identifier 10.1109/ACCESS.2017.2787798

Pattern Based Comprehensive Urdu Stemmer and Short Text Classification

MUBASHIR ALI, SHEHZAD KHALID, AND MUHAMMAD HASEEB ASLAM
Department of Computer Engineering, Bahria University , Islamabad, Pakistan.
Corresponding author: Mubashir Ali (mubbashircheema@gmail.com)

ABSTRACT Urdu language is used by approximately 200 million people for spoken and written communications. The bulk of unstructured Urdu textual data is available in the world. We can employ data mining techniques to extract useful information from such a large, potentially informative base data. There are many text processing systems available to process unstructured textual data. However, these systems are mostly language specific with the large proportion of systems applicable to English text. This is primarily due to language-dependent preprocessing systems, mainly the stemming requirement. Stemming is a vital preprocessing step in the text mining process and its primary aim is to reduce grammatical words form, e.g., parts of speech, gender, tense, and so on, to their root form. In the proposed work, we have developed a rule-based comprehensive stemming method for Urdu text. This proposed Urdu stemmer has the ability to generate the stem of Urdu words as well as loan words that belong to borrowed languages, such as Arabic, Persian, and Turkish, by removing prefix, infix, and suffix from the words. In the proposed stemming technique, we introduced six novel Urdu infix words classes and a minimum word length rule to generate the stem of Urdu text. In order to cope with the challenge of Urdu infix stemming, we have developed infix stripping rules for introduced infix words classes and generic stemming rules for prefix and suffix stemming. We also present a probabilistic classification approach to classify Urdu short text. Different experiments are performed to demonstrate the effectiveness and efficacy of the proposed approach. Comparison with existing state-of-the art approaches is also made. Stemming accuracy results demonstrate the adoptability of the proposed stemming approach for a variety text processing applications.

INDEX TERMS Infix classes, infix rules, stemming rules, stemming lists, Urdu stemmer, short text classification.

I. INTRODUCTION

There has been a development of exploration enthusiasm toward the advancement of textual data management techniques in the last few decades. These techniques are vital on the grounds that the amount of unstructured textual information is rapidly increasing with the remarkable development of electronic form of information such as internet and digital libraries, World Wide Web, electronic publications and documents. Text Mining is the extraction of previously obscure information from the cluttered text. Text mining tools are obliged to perform indexing, retrieval and recognition of this quickly developing text data.

Text classification is one of the important branch of textual data management techniques with its widespread application in variety of fields including automated news grouping and categorization, document organization, emotions categorization, financial analysis based on news, social website-based short text classification etc. One of the major applications of text classification is news classification. Long news is classified to great extent but work for news headline is specifically limited. News classification using complete news stories is computationally expensive. On the other hand, news headlines contain the crux of the news in a single sentence thus the computational processing of news headlines is efficient. There exists a variety of work that has performed headline-based news classification including financial news classification [1]–[7]. However, most of these systems are developed for English language and there is no significant work that has targeted Urdu language.

II. URDU LANGUAGE AND MORPHOLOGY

Urdu is the national language of Pakistan and state language of India. It is an Indo-Aryan language and is written from right to left. Urdu is widely spoken in India. Some states of India such as Delhi and Uttar Pradesh also use Urdu as their official language. According to an Indian

survey in 2011, 5% percent of Indian population speaks Urdu language. Approximately more than 200 million people use Urdu language as their communication medium, yet there is no effective text processing and automated categorization system for Urdu text. There is a dire need to have a comprehensive text processing system in Urdu language that can handle pre-processing requirements including stemming, stop words removal whilst presenting a good feature space representation and Urdu text categorization approaches.

Stemming is a fundamental pre-processing step preceding the tasks of text mining, information retrieval, and natural language processing. The primary goal behind the development of any stemmer is the augmenting of search effectiveness so the information retrieval systems can respond to users' query accurately. In linguistic morphology, stemming is a process of producing the stem/root form of the word by reducing its inflected or derived form. Urdu vocabulary is composed of many foreign languages i.e. English, Arabic, Persian, Turkish, Hindi, etc. The word 'Urdu' itself belongs to Turkish language. All these companion languages have their complex morphological structure. Due to robust morphology of borrowed languages, Urdu is an extremely rich morphological language which makes the stemming of Urdu language all that difficult. Urdu language is rich both in inflectional and derivational morphology [8]. Morphology is the study of internal structure of the words [9]. Inflectional morphology concerns with the grammatical formation of the words. Generating new words from the existing words is called derivational morphology. The major element of Urdu morphology is morpheme. Morpheme is a smallest language unit that holds some meaning. Morphemes are of two types i.e. free and bound morphemes [10]. The information retrieval system works on the base/root form of the words rather than the inflected or derived form. Therefore the development of an Urdu stemmer that has the ability to generate the stem of morphologically rich language is essential in order to boost the performance of IR and classification systems. Stemmer also has a vital role in the development of numerous interesting applications of natural language processing i.e. word parsing, spell checking, chunking, word sense disambiguation, statistical machine translation etc. Stemmer is an algorithm that generates the stem / root form of the word. Urdu stemmer produce the stem of a word by removing prefix, infix, and postfix attached to it, e.g. the stem of words خبروں (news), خبریں (news), اخبارات (newspapers), اخباروں (newspapers) and اخبار (newspaper) is خبر (news).

A. PROBLEMS IN DEVELOPMENT OF URDU STEMMER

There are some challenges pertaining to the development of stemmer for Urdu language. These challenges have been discussed in detail in [12]. Some of themajor issues are discussed in this section.

1) MORPHOLOGICAL FACTORS

As discussed earlier, Urdu is a rich morphological language. A single Urdu verb may have up to sixty different forms [33]. For instance, different forms of Urdu word لکھ (write) are: لکھا, لکھو, لکھے, لکھتی, لکھتا, لکھیں, لکھوا, لکھوائی, لکھوایا, etc. Also, a major part of Urdu vocabulary is composed of borrowed words from other languages such as Arabic, Persian, Turkish, etc. Intelligent handling of these borrowed words is a challenging task in the development of an effective Urdu stemmer.

2) INFIX STEMMING

An infix is a smallest unit that exists in the middle of any Urdu word e.g. in Urdu word بحاث (discussions), الف is an infix and the stem of بحاث is بحث (discussion). Most partof Urdu Language is influenced by Arabic grammar. Therefore, Urdu language has abundance of Arabic words having infixes such as انصاف (justice), مساجد (mosques), etc. Appropriate handling of these words can effectively improve the performance of IR system.

3) COMPOUND WORDS

The combination of two existing words is called a compound word. The correct identification of these words is extremely difficult as these combinations are not governed by specific rules. This problem becomes even more complicated for Urdu language as the words are coming from multiple languages. In [34] discusses three strategies of compound words formation inUrdu such as AB, A-o-B and A-e-B.

4) AB FORMATION

In AB formation only two free morphemes are joined together e.g. پٹی مرہم (bandaging), بیوی میاں (husband wife).

5) A-o-B FORMATION

In this formation, a joining morpheme is used between the words in order to make it free from morphemes. This linking morpheme is represented by a character ''و''. Examples of this compounds formation are عجزوانکساری (soberness and humility), امان وامن (law and order), etc.

6) A-e-B FORMATION

This formation of a word combines two different free morphemes with the help of one of the enclitic tiny morpheme; zer-e-izafat or hamza-e-izafat. For example صدرمملکت (president) is joined by adding a diacritic mark ''Zer'' below ر as zer-e-izafat. Similarly the word دل‌ءجذبہ (heart's spirit) is combined by inserting a diacritical mark hamza (ء) between جذبہ and دل as hamza-e-izafat. Reduplication of words also creates ambiguity during the stemming process; whether it is to be dealt as a single ora binary word.

Rest of the paper is organized as follow. Section III presents a brief review of existing text processing approaches in general as well as for Urdu text. A comprehensive Urdu stemmer is proposed in section IV. In section V, we present a probabilistic short text classification approach that employs stemmed text to perform news categorization. In Section VI, we present a methodology for enhancing conditional probabilities. Experiments are conducted to show the efficacy of the proposed Urdu stemming and classification approaches as compared to the competitors. These experiments are presented in section VII. The last section includes the conclusion and future work.

III. RELATED WORK

Automated text mining and recognition systems have gained significant attention in recent times. Considerable amount of work has been done in the development of text processing systems. However, these systems by and large are language specific due to the language dependent pre-processing systems mainly the stemming requirement. Most of the research has been carried out for processing English text. Comprehensive English stemmers, including Porter stemmer [12], are available for pre-processing English text. Similarly, wide range of work to perform high end processing of text including clustering and classification are available. Most of the existing work is focused on complete document processing and classification. However, it has soon been realized that processing of short text such as news headline can be extremely efficient whilst providing most of the desired information [1]–[7], [27]–[29]. However, there is little work available for processing of short text in Urdu language. There is a dire need of the development of comprehensive Urdu stemmer to enable variety of text processing systems effectively applicable to Urdu text.

Stemming can be performed by using three common approaches i.e. affix stripping, table lookup, and statistical methods [11]. Affix removing approach depends on the morphological structure of the given language. This approach is used to obtain the stem of the word by removing the attached prefix and postfix from the word. A well-known porter stemmer for English language is an example of this approach [12]. In table lookup approach each word and its associated stem is stored in structured table. This approach requires a lot of storage space for its implementation and its table needs to be updated manually for each new word. In Statistical approach, based on the size of corpus words formation rules are developed. Some methodologies are used i.e. frequency count, n-gram [13], Hidden Markov Models [14], and link analysis [15]. Until now, lots of stemming methods [12], [16]–[23] have been proposed for a number of languages i.e. English [8], [16]–[19], Arabic [20], [21], Persian [22], [23] etc. These stemming methods are based on rule based strategy. In literature, there exists many stemming methods [13], [14], [24] that are developed by using statistical approach. Rule based approaches are highly dependent on the deep morphological knowledge of the language, whereas statistical analysis is performed on the basis of corpus size. The study [16] developed the first stemming method for English language. This stemming approach is based on rule based strategy and comprises of 260 stemming rules. This stemming method generates the stem of English word in two phases. In the first phase of the stemmer, the maximum matched suffix is removed (defined in suffix table) and the word is recoded to generate suitable stem. Spelling exclusions are covered in the second phase of the stemmer. This stemmer is known as Lovins stemmer. John [17] came up with another rule based stemming method. It is an extension of J.B. Lovins stemmer and covers a comprehensive list of 1200 suffixes. The suffixes are stored in reversed order listed by their length and last character. This method covers more suffixes than Lovins stemmer. Porter in 1980 [12], [18] developed a rule based stemmer for English language. He simplified the rules of Lovins stemmer to about 60 rules. In this proposed stemming method, suffixes are removed from words by using suffix list and some conditions are enforced to find out the suffixes to be de-attached. This is one of the most popular stemming methods for English textual data and is known as Porter stemming algorithm. Porter also designed a stemming framework referred to as ''snowball''. The objective behind the development of this framework is to allow the programmer to develop their own stemmer for languages. Porter [12], [18] discovers the problems of over-stemming, under-stemming, and mis-stemming. Paice [19] came up with another stemming method based on rule-based strategy. It is an iterative algorithm based on a table comprising 120 rules that are indexed by the last letter of a suffix. In each iteration, it tries to find an appropriate rule by the last character of the word. Each rule is used either for deletion or replacement of an ending. If none of the rule is found, it terminates.

In past, many stemming algorithms have been developed for South Asian languages. Khoja and Garside [20] developed a superior root-based stemming method for Arabic language. This stemming method generates the stem of an Arabic word by removing prefix, infix, suffix, and then uses pattern matching. In order to improve the stemming accuracy of proposed stemming approach, this stemmer uses several linguistic data files i.e. punctuation character, diacritic characters, and a list of 168 stop words. For Arabic text, Thabet [21] proposed a light stemming approach. It is developed by using rule based approach and is applied on classical Arabic in Quran. This Arabic stemmer generates the list of words from each surah. If the word in list is not found in the stop word list then prefix is truncated from the word. Stemming accuracy of proposed algorithm for prefix stemming is 99.6% and 97% for postfix stemming. Tashakori *et al.* [22] came up with first Persian stemmer called Bon that is based on rule-based approach. It is an iterative matching algorithm that removes all the possible affix and suffix from the word until required. After truncation of prefixes and suffixes a re-coding technique is used to generate the valid stem. With the use of Bon, recall is improved by 40%. Mokhtaripour and Jahanpour [23] developed another stemming method for Persian language by using rule based strategy. This stemmer generates the stem of Persian text without using language dictionary. The performance of a query system was improved up to 46% by using this developed stemmer.

As far as Urdu language is concerned very few stemming methods have been proposed i.e. Asass-band [8], Light Weight stemmer for Urdu text [25], Novel Urdu stemmer [30], Template based affix striping Urdu stemmer [31] and Rule based Urdu stemmer [32]. These [8], [25] stemming methods generate the stem by removing pre-fix and postfix present in the Urdu words. Both these stemmers are highly dependent on very large rules lists as well as exception lists. These large lists significantly affect the efficiency of these Urdu stemmers. As Urdu language is composed of many foreign languages such as English, Arabic, Persian, Turkish, etc. both of these [8], [25] existing stemming techniques are unable to generate the stem of words that belong to borrowed languages. In Urdu morphology there are many words that have infix in it in addition to prefix and postfix. The truncation of infix from Urdu words is crucial for an effective Urdu stemmer. Khan *et al.* [32] have proposed some Urdu infix classes and developed rules

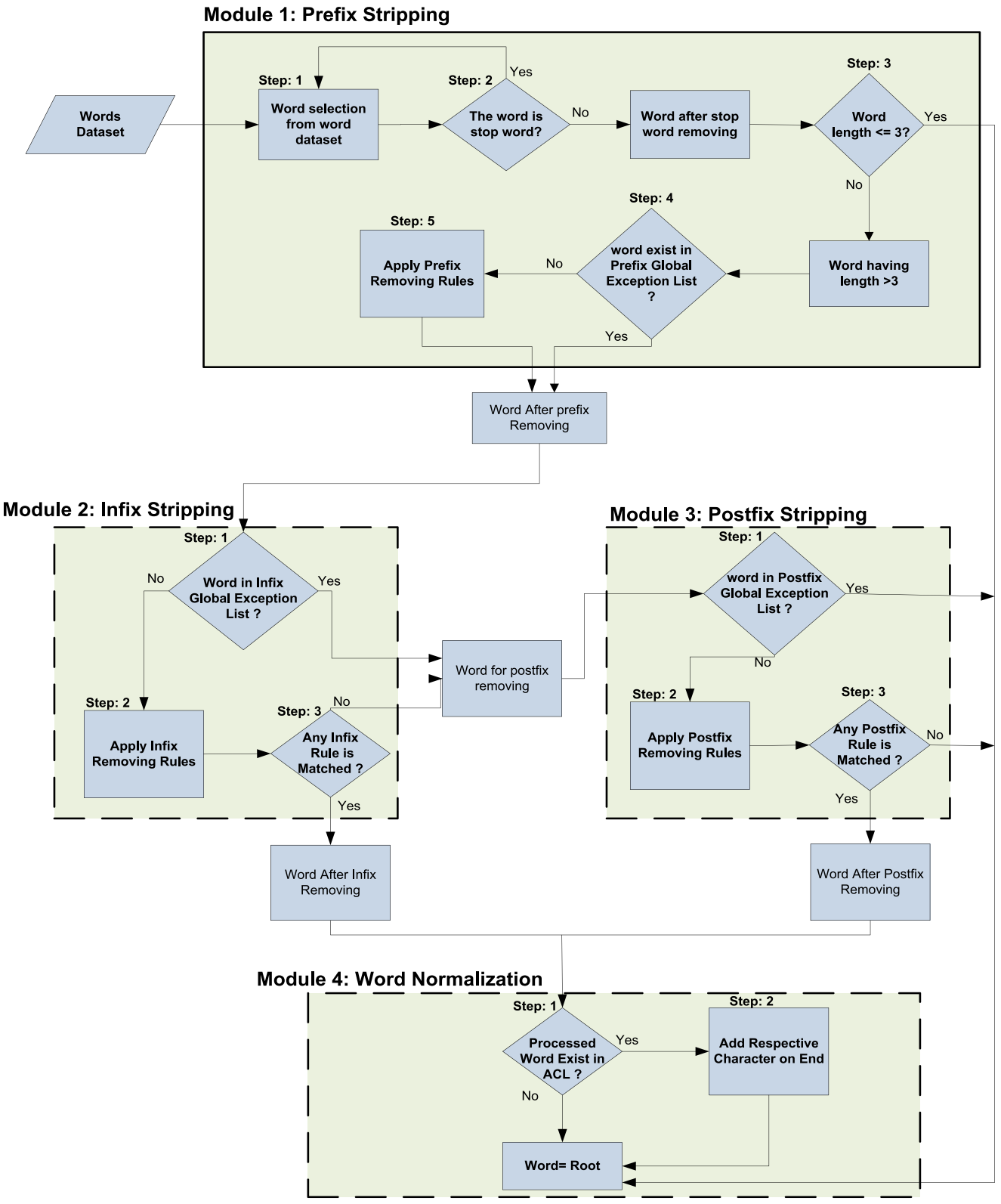

FIGURE 1. Flow diagram for proposed system.

for all the classes to deal with the challenges of Urdu infix stemming. This approach established more accuracy in comparison to [31].

IV. PROPOSED URDU STEMMER

The development of an effective Urdu stemmer is considered to be a challenging task in text mining research community.

This is mainly due to complex morphological structure of Urdu language. To cope with this challenge, we have proposed a rule based Urdu stemmer. A complete overview of the proposed Urdu stemmer system is given in Figure 1. All stemming rules (prefix, infix, and suffix) and stemming lists i.e. Prefix Global Exception List (PrGEL), Infix, Infix Global Exception List (InGEL) and Postfix Global Exception List (PoGEL) used for the development of proposed Urdu stemmer are explained with detail in section A and section B. Proposed stemmer is composed of four different modules. In module 1, less informative words are eliminated from the dataset and prefix stripping is performed on a given word. After that, the preprocessed word is passed to module 2 to perform infix stripping on it. The infix stripped word is then used in module 3 for the removal of postfix. At the end, module 4 is used to get the stem form of a word. Module 1 of prefix stripping is composed of five different steps. In step 1, a word is selected from word dataset. Selected word is compared with a static list of less informative word in step 2. The word is filtered in step 2 if its match is found in the list of less informative word. Step 3 checks whether the word itself is a root by looking at the word's length. Step 4 performs matching of the word with Prefix Global Exception List (PrGEL). Prefix removing rules are applied in step 5 if the word does not exist in PrGEL. The word pre-processed by module 1 is then passed to module 2 to perform infix stripping to reduce the word to its stem. This module is comprised of three various steps. In step 1, the word is searched in Infix Global Exception List (InGEL). The remaining infix stripping steps are ignored if the match of word is found in InGEL and the word is moved to module 3 for postfix stripping. If not the infix removing rules are applied at step 2. If any one of the infix rule is matched/applied, the processed word is moved to module 4 in order to generate the surface form of word. On the other hand, if no rule is matched the word is moved to module 3 for postfix stripping. After prefix and infix stripping are performed, the word is used in module 3 for postfix stripping. Module 3 of postfix stripping is comprised of three steps. In step 1, the word is filtered out and marked as a stem word, if it is found in Postfix Global Exception List (PoGEL). Postfix removing rules in step 2 are applied and maximum matched suffix are removed from the word if it does not exist in PoGEL. If a rule-match is found and the word is processed, it is moved to module 4 to normalize it. One the other hand, if no rule-match is found the word is considered to be reduced to its stem. Module 4 is used to produce the surface form of the word if required. If the processed word from module 2 and module 3 exists in Add Character Lists (ACLs), corresponding characters are attached to the end of word. Otherwise, the word is considered as a stem word.

Our proposed stemmer is centered on the rule-based affix stripping approach to generate the stem of Urdu as well as borrowed words. This Urdu stemming approach is comprised of various infix words classes, stemming rules, stemming list, and stem word dictionary.

A. STEMMING RULES

In this stemmer, we have developed three kinds of stemming rules i.e. prefix, infix, and postfix rules. The existing state-of-the art approaches [8], [25] have also developed prefix and postfix stemming rules, but they presented a vast set of rules. In this stemming work, we have minimized existing stemming rules and proposed generic rules that are applicable to any type of Urdu words. Our developed rules are also capable of producing the stem of borrowed words.

1) MINIMUM WORD LENGTH RULE

After a detailed analysis of Urdu morphology, it is observed that an Urdu word that comprises of only two or three characters is already a stem word. For example, the words دن (day), رات (night), وقت (time) are already stemmed words. These words are treated as stem words and filtered out to avoid further stemming processing. The finding of this rule is a novel contribution of proposed Urdu stemmer. Some example words of this rule are given in Table-1.

TABLE 1. Examples of words handled by minimum word length rule.

دن (Day)	رات (Night)
وقت (Time)	صبح (Morning)
قبر (Grave)	سفر (Travel)

2) PREFIX REMOVING RULES

Prefix is a morpheme that is attached to the beginning of the word. In Urdu morphology it is known as سابقہ. The prefix may compose of one or two characters and sometimes it is a complete word. In order to develop prefix stripping rules, various grammar books and Urdu literature is studied. After this profound study a list of 60 prefixes is generated. These stemming rules are rationally small in number as compared to rules developed in existing state-of-the art techniques [8], [25]. Some prefix stripping rules are presented in Table-2.

TABLE 2. Examples of prefix stripping rule.

ال	بر
در	زي
از	نا

3) INFIX REMOVING RULES

The most prominent work of this stemming method is infix stripping. Most part of Urdu grammar is influenced by Arabic grammar. Therefore, Urdu morphology has inherited features of this parent language. To cope with the challenges of Urdu infix stemming, we consulted linguistic experts with strong command on Urdu morphology who devised Urdu infix classes comprised of words that contain infix. Infix rules were then applied to OR developed against each Urdu infix class in order to remove infix attached to words. These developed rules applied on Urdu words and generated stem of words is validated from linguistic experts and from online Urdu dictionary [35].

Developed Urdu infix classes are Alif Arabic Masdar (infinitive verbs beginning with Alif), Te Arabic Masdar (Infinitive verbs beginning with Te), Isam Fiale (Active subject), Isam Mafool (passive object), Arabic Jamah (Arabicplural words), and Isam Zarf Makaan (place showing noun).

In order to identify theArabic words for applying proposed infix rules, the characters i.e. (instances of prefix rules are presented in Table-2. (کہ گہ چہ ٹہ تہ گ ژ ڑ ڈ چ ٹ پ) are verified in the Urdu word. If any one of the character is found in the Urdu word then this word is not considered to be a part of Arabic language. To remove infixes from words that belong to proposed Arabic infix classes, we have developed variety of .infix rules in this section. These rules are grouped w.r.t infix classes that they handle. All infix generated rules are given in tabular form in Appendix A.

a: ALIF ARABIC MASDAR (INFINITIVE VERBS BEGINNING WITH ALIF) CLASS INFIX STRIPPING RULES

In order to remove the infixes of this class, we have developed the following rules:

Rule-1: If a word starts with Alif (''الف'') and the length of word is exactly equal to five, then remove all the Alif (''الف'') from this word.

Samples of this rule are given in Table 3.

TABLE 3. Example of words handles by proposed alif massdar infix word class.

Rule No	Original Word	Stem Word	Original Word	Stem Word	Original Word	Stem Word
Rule -1	اسناد Certif) (icates	سند Certif) (icate	اخبار (Newspa pers)	خبر (New s)	اوصاف Quali) (ties	وصف Quali) (ty
Rule -2	استقبال (Recep tion)	قبل Befor) (e	اخلاقیات (Moralit ies)	خلق (Crea tion)	انتشار (Broa dcast)	نشر Broa) (dcast
Rule -3	اتباع (Follo wers)	تبع (Follo w)	اتحاد Allianc) (e	تحد (Alli ance)	اتحاف Gifts) (	تحف (Gift)
Rule -4	احسانات (Blessi ngs)	حسن Bless) (	احسانوں Blessin) (gs	حسن Bles) (s	افسادی (Quar rels)	فسد (Quar rel)
Rule -5	احتساب(Accou ntabilit y)	حسب (Acco unt)	احتسابی (Accoun tability)	حسب (Acc ount)	ابتسام (Open ing)	بسم (Open)

Rule-2: If a word starts with Alif (''الف'') and the length of word is greater than five, then remove all the Alif (''الف'') Te (''ت'') Sin (''س'') ChhotiYeh (''ى'') NunGunna (''ں'') ChhotiYeh hamza (''ۓ'') Waohamza (''ؤ'') andhamza (''ء'') from this word.

Examples of this rule are shown in Table 3.

Rule-3: If a word starts with Alif (''الف'') and the character at index one is Te (''ت'') and length of the word exactly equals to five, then remove all the Alif (''الف'') ChhotiYeh (''ي'') Nun Gunna (''ں'') ChhotiYeh hamza (''ۓ'') Waohamza(''ؤ'') hamza (''ء'') and Wao (''و'') from this word.

Examples of this rule are presented in Table 3.

Rule-4: If a word starts with Alif (''الف'') and the character at index two is Sin (''س'') and length of the word is exactly greater than five, then remove all the Alif (''الف'') Te (''ت'') ChhotiYeh (''ي'') NunGunna (''ں'') ChhotiYeh hamza (''ۓ'') Waohamza (''ؤ'') hamza (''ء'') Wao (''و'') do-chashmihe (''ہ'') and BadiYeh (''ے'') from this word.

Example words of this rule are presented in Table 3.

Rule-5: If a word starts with Alif (''الف'') and the character at index three is Sin (''س'') and length of the word is exactly greater than five, Then remove all the Alif (''الف'') Te (''ت'') ChhotiYeh (''ي'') NunGunna (''ں'') ChhotiYeh hamza (''ۓ'') Waohamza (''ؤ'') hamza (''ء'') Wao (''و'') do-chashmi he (''ہ'') and BadiYeh (''ے'') from this word.

Samples of this rule are presented in Table 3.

b: TE ARABIC MASDAR (INFINITIVE VERBS BEGINNING WITH TE) CLASS INFIX STRIPPING RULES

To remove infixes from the words related to this class, following rules are developed:

Rule-1: If a word starts with Te (''ت'') and also contain Alif (''الف''), then remove all the Alif (''الف''), Te (''ت''), Nun Gunna (''ں''), ChhotiYeh (''ي'') and BadiYeh (''ے'') from the word.

Samples of this rule are given in Table 4.

Rule-2: If a word starts with Te (''ت'') and length of the word is exactly equal to five and second last character of the word is ChhotiYeh (''ي'') then remove all the Te (''ت''), Nun Gunna (''ں''), and BadiYeh (''ے'') from this word.

Examples of this rule are given in Table 4.

TABLE 4. Example of word handled by proposed TE arabic masdar infix word class.

Rule No	Original Word	Stem Word	Original Word	Stem Word	Original Word	Stem Word
Rule -1	تساہل (Respite)	سہل (Easy)	تشاکل (Shapes)	شکل Shape) (	تعداد Numbers) (	عدد Numbe) (r
Rule -1	تعاقب (Pursuit)	عقب (Pursu e)	تعارض (Request)	عرض Reque) (st	تعارف (Recognit ion)	عرف (Recogn ize)
Rule -2	تصدیق (Verifica tion)	صدق (Verif y)	تحسین (Admirati on)	حسن (Admi re)	تعلیم Educatio) (n	علم (Educat e)
Rule -2	تحریم (Respect able)	حرم (Respe ct)	تحصیل (Achieve ment)	حصل (Achie ve)	تکریم (Kindnes s)	کرم (Kind)

c: ISAM FIALE (ACTIVE SUBJECT) CLASS INFIX STRIPPING RULES

In order to remove the infixes of this class, we have developed the following rules:

Rule-1: If word length is exactly equal to four and also contain Alif (''الف'') then remove all the Alif (''الف'') from this word.

Some example words of this rule are presented in Table 5.

Rule-2: If word length is exactly equal to four and the 2nd last character of the word is ChhotiYeh (''ي''), then remove all the ChhotiYeh (''ي'') from this word.

Some example words of this rule are presented in Table 5.

TABLE 5. Example of word handles by proposed Isamfiale infix word class.

Rule No	Original Word	Stem Word	Original Word	Stem Word	Original Word	Stem Word
Rule -1	قادر (Powerful)	قدر (Power)	ساجد (Bower)	سجد (Bow)	شاہد (Witness)	شہد (Wit)
Rule -1	واجد (Devotion)	وجد (Devote)	شاکر (Thankful)	شکر (Thank)	کاشف (Revealers)	کشف (Reveal)
Rule -2	قدیر (Powerful)	قدر (Power)	رقیب (Competitor)	رقب (Compete)	شریف (Respectable)	شرف (Respect)
Rule -2	رشید (Guidance)	رشد (Guide)	صغیر (Littleness)	صغر (Little)	تعزیم (Greatness)	عزم (Great)

d: ISAM MAFOOL (PASSIVE OBJECT) CLASS INFIX STRIPPING RULES ISAM MAFOOL (PASSIVE OBJECT) BEGINNING WITH MEEM 'م'

To remove infixes from the words related to this class, following rules are developed:

Rule: If a word starts with Meem ("م") and length of the word is exactly equal to five and 2nd last character of the word is Wao ("و") then remove all the Wao ("و") and Meem ("م") from this word.

Example words handled by this rule are given in Table 6.

TABLE 6. Examples of words handled by proposed Meem ("م") isammasool infix rule.

Original Word	Stemmed Word
منظور (Approved)	نظر (Approve)
مجذوب (Merger)	جذب (Merge)
مجبور (Depressed)	جبر (Depress)
محصول (Achievements)	حصل (Achieve)
محکوم (Subordinate)	حکم (Order)

e: ARABIC JAMAH (ARABIC PLURAL) CLASS INFIX STRIPPING RULES

To remove infixes from the words related to this class, following rules are developed:

Rule: If word length is exactly equal to four and 2nd last character of the word is Wao ("و") then remove Wao ("و") from this word.

Examples words handled by this rule are given in Table 7.

TABLE 7. Examples of word handled by proposed arabic Jamah infix rule.

Original Word	Stemmed Word
قبور (Graves)	قبر (Grave)
سدور *(Presidents)*	سدر *(President)*
شکور *(Thankful)*	شکر *(Thank)*
قدوس *(Piousness)*	قدس *(Pious)*

f: ARABIC JAMAH AND ISAM FIALE (ARABIC PLURALS AND ACTIVE SUBJECT) BEGINNING WITH MEEM ("م")

In order to remove infixes from the words related to this class, following rules are developed:

Rule-1: If a word starts with Meem ("م") and it also contains Alif ("الف") then remove all the Alif ("الف"), Te ("ت"), Noon Ghuna ("ں"), ChhotiYeh ("ۓ"), BadiYeh ("ے"), and do-chashmi he ("ہ") from this word.

Examples words handled by this rule are given in Table 8.

TABLE 8. Example of words handled by proposed arabic Jamah and Isamfiale beginning with Meem ("م") infix rule.

Rule No	Original Word	Stem Word	Original Word	Stem Word	Original Word	Stem Word
Rule-1	مناسبت (Dedication)	نسب (Dedicate)	متبادر (Lighter)	بدر (Light)	متعارض (Requester)	عرض (Request)
Rule-1	مجاہد (Struggler)	جہد (Struggle)	محافظ(Protector)	حفظ(Protect)	منافقت (Concealer)	نفق (Conceal)
Rule-2	منتظر (Waiter)	نظر(Wait)	منتسب (Dedication)	نسب (Dedicate)	منتظم (Manager)	نظم (Manage)
Rule-2	منتقل (Moveable)	نقل (Move)	منتفع (Profits)	نفع (Profit)	منشور(Declaration)	نشر (Declare)

Rule-2: If a word starts with Meem ("م") and the character at index two is Te ("ت") and length of the word is exactly equal to five, then remove all the Meem ("م"), Te ("ت"), Noon Ghuna ("ں"), ChhotiYeh ("ۓ") from this word.

Examples words handled by this rule are given in Table 8.

4) POSTFIX REMOVING RULES

Postfix is a morpheme that is attached at the end of the word. In Urdu morphology it is known as (لاحقہ). The postfix may consist of one or two characters and sometimes may be a complete word. A list of 140 suffixes is generated after a deep study of Urdu grammar and literature books.

Examples of these suffixes are presented in Table-9.

TABLE 9. Examples of postfix removing rule.

وں	وار
یں	ویں
واں	نیں

B. STEMMING LISTS

In order to develop proposed Urdu stemmer, we have developed some stemming lists i.e. prefix global exception lists, infix global exception list, postfix globalexception list, stop words/less informative words list, Stem word dictionary, and add character list.

1) PREFIX GLOBAL EXCEPTION LIST (PrGEL)

Urdu language is rich in morphology, so it is important to correctly identify the prefix from Urdu words. The misunderstanding of the prefixes can most likely lead to poor stemming

results as well as the loss of useful words. Urdu morphology contains certain words that have prefixes attached to them which can not to be detached because they are a part of those words. For example if the prefix "با" is removed from the word "بارش" (rain) then it produces "رش" which is not an Urdu word. On the other hand we cannot remove the prefix "با" from the prefix rules list because this prefix generates the stem ofmany other important words. Therefore such words should be treated as exceptional cases in order to keep the meanings of words intact. In this proposed Urdu stemmer, we have developed an exception list of about 5000 words that is significantly smaller in size as compared to the lists of existing stemming state-of-the art technique [8], [25]. Some sample words of this list are given in Appendix B.

2) INFIX GLOBAL EXCEPTION LIST (InGEL)

Urdu morphology is influenced by Arabic grammar so there are many words in Urdu morphology that are from Arabic morphology and also contain infixes. For example the words "اتوار" (Sunday), and "الماری" (wardrobe) have infixes attached to them. But these infixes are a part of word and cannot be removed. During the formulation of infix stripping rules these words are identified. In order to preserve the meaning of these words they must be known in advance and handled as an exceptional case. In this purposed stemming work we have developed a list of 3000 words that is known as infix global exception list. Example words of this exception list are presented in Appendix B.

3) POSTFIX GLOBAL EXCEPTION LIST (PoGEL)

Similar to prefix identification, the correct recognition of postfix is essential for effective stemming work. During the execution of postfix rules, when a postfix is removed from the word, may be an invalid stem of the word is generated. This is due to the irrelevant truncation of the postfix. For example, in the word "ہاتھی" (elephant) when suffix "ی" is removed then it produces the stem "ہاتھ" (hand), which is unacceptable OR that changes the meaning of the word. In order to maintain the originality of such words, an exception list of about 6000words has been generated. This list is known as postfix global exception list. Some samples of this exception list are given in Appendix B.

4) STOP WORDS/LESS INFORMATIVE WORDS LIST

In Urdu text there are many words that occur frequently but they do not contribute in the Urdu text mining process, such words are known as Stop Words. In order to filter out these less informative words from Urdu text, a static list of 200 words is generated. This list is generated after consulting various grammar books and Urdu literature. Some example words are given in Table-10.

TABLE 10. Examples of less informative words.

کا	ہم
تم	وہ
کو	ان

5) STEM WORD DICTIONARY

To check the stemming accuracy of the proposed Urdu stemmer, we have developed a generic stem word dictionary of about 10000words. Every stem generated by the proposed stemming rules is validated by using this stem word dictionary. This stem dictionary is developed after a detailed study of Urdu morphology. Some instances of stem words are presented in Table-11.

TABLE 11. Examples of stem words.

نظر *(Viewing)*	نسب *(Dedicate)*
جذب *(Merge)*	بدر *(Light)*
جبر *(Harsh)*	عرض *(Request)*

6) ADD CHARACTER LISTS (ACLs)

In some cases, the execution of proposed infix and postfix rules generate an incomplete stem of the word. For example after stripping the postfix from the word "جگہوں" (places), we get "جگ" which is an incorrect stem. To produce the correct stem i.e. "جگہ" (place) of the word "جگہوں" (places), a character Hey "ہ" should be added at the end of the word "جگ". In order to generate a meaningful stem, we have developed eight different types of lists for characters (ی،ہ،و،ں،س،ر،ت،ک،الف). Execution of these rules is presented in Appendix C.

C. PROPOSED URDU STEMMER ALGORITHM

The proposed Urdu stemmer algorithm works on the basis of longest match theory. This theory states that when more than one affixes rules are matched for a word, then the longest match affix should be removed. Therefore, it is necessary to find out all possible matched affixes rather than removing the immediately matched affix. Our proposed stemmer evaluates all the possible matching affixes at once and arranges them based on their length.

The algorithm is comprised of the following steps:

a) Select a word from the dataset.
b) Filter out the word if it is a stop word such as if its match is found in the non-informative word list. Ignore that word and select the next one from the word sequence.
c) Determine the length of selected word.
   i. If the length of word is less than or equal to three, mark the word as a stem word and go to (g).
   ii. If the word length is greater than three, go to (d).
d) Search the word in Prefix Global Exception (PrGEL) List.
   i. If word exists in PrGEL then go to (e).
   ii. If word does not exist in PrGEL, then apply prefix removing rules and remove the maximum matched prefix from the word and go to (e).
e) Search the word in Infix Global Exception (InGEL) List.
   i. If the word is found in InGEL, then go to (f).
   ii. If the word is not found in InGEL, then apply the infix removing rules.

iii. If any one of the infix rule is applied, search the processed word in Add Character Lists (ACLs).
iv. If processed word is discovered in any ACLs, then attach the respective character to the end of processed word. Mark the processed word as stem and go to (g).
v. If processed word does not exist in any ACLs, mark the processed word as stem and go to (g).
vi. If none of the infix rules is applied, go to (f).

f) Search the word in Postfix Global Exception (PoGEL) List.
i. If the word found in PoGEL, mark the processed word as stem and go to (g).
ii. If the word does not exist in PoGEL, then apply the postfix removing rules.
iii. If any one of the postfix removing rule is matched, then remove the maximum attached suffix from the word and search the processed word in Add Character Lists (ACLs).
iv. If the processed word is found in any ACLs, then attach the respective character to the end of processed word. Mark the processed word as stem and go to (g).
v. If processed word is not found in any ACLs, mark the processed word as stem and go to (g).
vi. If none of the postfix rule is applied then mark the word as stem and go to (g).

g) Repeat from a-f for all words.

V. SHORT TEXT CLASSIFICATION

In this section, we present our short text classification approach, employing the comprehensive stemmer as presented in section IV. Let $\mathbf{DB}_{\text{train}}$ be the complete set of labeled news headlines belonging to various news categories $\mathbf{C}$, we identify unique words list $\mathbf{UW}$ as:

$$\mathbf{UW} = \left\{ \mathbf{UW} \cup \tilde{W}_i \; \textit{if} \; \tilde{W}_i \notin \mathbf{UW} \wedge \mathrm{CL}(W_i) \geq 3 \wedge W_i \notin \mathbf{SW} \right\} \quad \forall W_i \in \boldsymbol{DB}_{train} \tag{1}$$

where $CL(.)$ is a function that returns the number of characters in a given word, $\mathbf{SW}$ is a stop word list and $\tilde{W}_i$ is the stemmed form of Wi obtained by applying the stemming algorithm as presented in section IV. After the identification of unique words list $\mathbf{UW}$, we then compute histogram of occurrence of each word from $\mathbf{UW}$ in various headlines belonging to different news headline categories. Let M be the total number of words in unique words list $\mathbf{UW}$ and N be the total number of news categories, the histogram $\boldsymbol{H}$ is a 2D matrix of $N \times M$ dimensions where $H(i, j)$ represents the occurrence of word Wi $\in \mathbf{UW}$ in various news headlines belonging to class $C_j \in \mathbf{C}$.

The conditional probability $P(Cj|Wi)$ of any class C_j given a word Wi is computed as:

$$P(C_i \mid W_i) = \frac{H_{(i,j)}}{\sum_{j=1}^{N} H_{(i,j)}} \quad \forall i, j \tag{2}$$

Once the probabilistic model of the known news categories are learnt, unseen Urdu news headline L is classified by computing the probability of each class Cj and assigning L to the class with maximum probability. This is achieved by preprocessing the words in L as:

$$\mathbf{W_L} = \left\{ \mathbf{W_L} \cup \tilde{W}_i \; \textit{if} \; \tilde{W}_i \in \mathbf{UW} \wedge \mathrm{CL}(W_i) \geq \mathbf{3} \wedge W_i \notin \mathbf{SW} \right\} \quad \forall W_i \in \mathbf{L} \tag{3}$$

The probability of various classes given as $\mathbf{W_L}$, using weighted probabilistic framework, is then computed as:

$$P(C_j) = \sum_{i=1}^{|W_L|} \sqrt{P(C_j|W_i)} \tag{4}$$

The Urdu news headline is then classified to the news category C_k with the maximum probability, identified as:

$$k = \arg min_j P(Cj) \quad \forall j \tag{5}$$

VI. ENHANCING CONDITIONAL PROBABILITIES

In this section, we present our approach to enhance the performance of probabilistic classifier. We propose to penalize the conditional probabilities $P(C_j|W_i)$ given the news L containing word W_i is misclassified to C_j. The proposed approach is an iterative algorithm to enhance conditional probabilities with the aim to minimize misclassification. The algorithm for evolving conditional probabilities comprises of the following steps:

1. Initialize all the weights in the weight matrix (ω) of size (N×M) by 0.
2. Select Urdu news headline L from training dataset $\boldsymbol{DB}_{train}$ and extract the set of valid stemmed words as specified in eq. (1).
3. Classify the news headline L using eq. (4). Let C_{gt} be the actual category of L obtained from the ground truth and C_j be the category identified by the proposed classifier, the weights corresponding to the words in L is then updated as:

$$\omega_{i,j} = \begin{cases} \omega_{i,j} + 1 & \textit{if } C_{gt} = C_j \\ \omega_{i,j} & \textit{otherwise} \end{cases} \quad \forall W_i \in L) \tag{6}$$

4. Repeat steps 2-3 for all headlines in $\boldsymbol{DB}_{train}$
5. Normalize the weights as:

$$\omega_{i,j} = \frac{\omega_{i,j}}{\sum_{j=1}^{N} \omega_{i,j}} \quad \forall i, j \tag{7}$$

6. Penalize the learned conditional probabilities based on the weight matrix, representing the misclassification contributions, as:

$$\mathrm{P(C}_i|\mathrm{W}_i = \mathrm{P(C}_i|\mathrm{W}_i) - \alpha(t) \times (\omega_{i,j} \times \mathrm{P}(\mathrm{C}_i|\mathrm{W}_i)) \tag{8}$$

Where α(t) is the adaptation rate that controls the fraction of change to be made in the existing conditional probabilities.

7. Normalize the updated conditional probabilities using:

$$P(C_i|W_i) = \frac{P(C_i|W_i)}{\sum_{j=1}^{N} P(C_i|W_i)} \quad \forall i, j \tag{9}$$

8. We shift from coarse adjustment to fine adjustment by changing the learning rate exponentially over time as:

$$\alpha\,(t) = 1 - e^{\frac{2(t-t_{total})}{t_{total}}} \quad (10)$$

where t_{total} is the total number of adaptation iterations.

9. Repeat steps 1-7 for t_{total} iterations.

VII. EXPERIMENTAL EVALUATION

In this section, we demonstrate the effectiveness of proposed approach for Urdu text pre-processing and classification of short Urdu text.

A. EXPERIMENTAL DATASETS

Experiments are conducted on 4 corpuses. A brief overview of these Urdu corpuses is presented in Table-12.

TABLE 12. A brief overview of experimental datasets.

Corpora	Description	Total Words	Unique Words
Corpus 1 (C1)	An Urdu headline news corpus. It contains the news of two different categories i.e. politics and weather.	12500	5070
Corpus 2 (C2)	It is also an Urdu headline news corpus. It comprises of two different news classes i.e. sports and terrorist.	7250	3080
Corpus 3 (C3)	It consists of unique Urdu word. It has developed by using various grammar books and Urdu dictionaries.	24238	24238
Corpus 4 (C4)	A comprehensive headline news corpus obtained by combining corpus 1, corpus 2 and corpus 3.	43988	32388

B. EXPERIMENT 1: EVALUATION OF PROPOSED URDU STEMMER

The purpose of this experiment is to evaluate the performance of proposed Urdu stemming algorithm using variety of corpus. In order to evaluate the stemming accuracy of proposed stemming rules, experimental datasets are filtered out from diacritic, special symbols and less informative words in pre-processing step. After the pre-processing steps, 32000 unique words are extracted. We perform the evaluations for each rule separately to analyze its contribution in the overall stemming framework.

TABLE 13. Words handled by proposed minimum word length rule.

Corpora	Words Having Length !=3
Corpus 1	351
Corpus 2	221
Corpus 3	4380
Corpus 4	4952

1) EVALUATION OF PROPOSED MINIMUM WORD LENGTH RULE

In order to evaluate the effectiveness of minimum word length rule, it is applied on the pre-processed experimental datasets. The accuracy results of this stemming rule are shown in Table-13. As evident from the results of this rule, the proposed minimum word length rule successfully detects the words which are stems themselves. This rule avoids further application of prefix, infix and postfix stemming rules which may even destroy the word whilst increasing the computational complexity.

TABLE 14. Stemming accuracy of proposed prefix rules.

Corpora	Total Words Tested	Number of Words that Matched Prefix Rules	True Positive	False Positive	Accuracy %
Corpus 1	4468	195	167	28	85.64%
Corpus 2	2722	182	160	22	87.91%
Corpus 3	19858	323	270	53	83.59%
Corpus 4	27048	700	597	103	85.28%

2) EVALUATION OF PROPOSED PREFIX RULE

After the application of minimum word length rule, we extract 27048 words for the rest of the stemming process. The effectiveness of stemming rules i.e. prefix, infix, and postfix rules are evaluated by using the number of words that matched stemming rules. To elaborate the results, we also calculate the true positive (correctly stemmed words) and false positive (incorrectly stemmed words) against every stemming rule. The stemming accuracy of proposed Urdu stemmer is calculated as the ratio of true positive and the number of words that matched stemming rules. The results produced by the application of proposed prefix rules for each corpus are given in Table-14. Stemming accuracy results of proposed prefix rules w.r.t. precision, recall, and f1 measure are given in Table-15 and figure-2 presenting the visualized results.

3) EVALUATION OF PROPOSED INFIX RULE

After the application of prefix stripping rules, we used prefix-stripped words for the evaluation of proposed infix stripping rules. Table-16 shows i) The stemming accuracy results of each of the proposed infix word class with its associated infix rules, ii) the results produced by the application of infix rules with respect to each infix class, iii) and the effectiveness and adoptability of the proposed rules. Table-17 shows the

TABLE 15. Sumarized stemming results of proposed prefix rules.

Stemming Characteristics	Corpus
Words consider for prefix stemming	27048
Words already free from prefix's	26348
Prefix to be removed from words	700
Correctly removed prefix's	597
Proposed Stemming Accuracy Results	
Precision	85.28%
Recall	99.6%
F1-Measure	91.8%

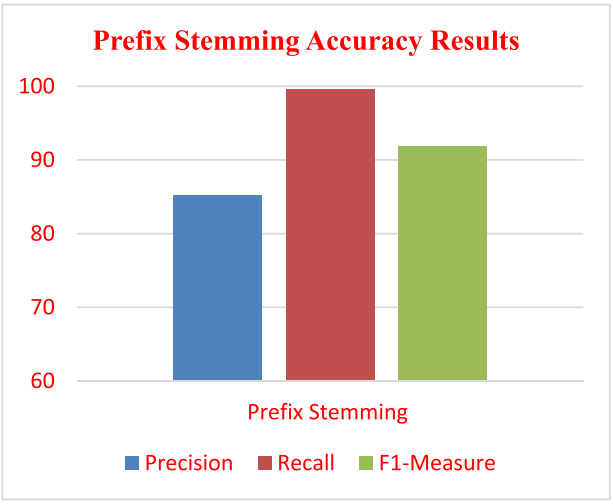

FIGURE 2. Visualized results of prefix stemming.

TABLE 16. Stemming accuracy results of proposed infix stripping rules.

Infix Word Class	Corpora	Total Words Tested	Number of Words that Matched Prefix Rules	True Positive	False Positive	Accuracy
Alif Arabic Masdar	Corpus 4	27048	4300	3679	621	85.55%
Te Arabic Masdar	Corpus 4	27048	2815	2563	252	91.04%
Isam Fiale	Corpus 4	27048	6783	6021	762	88.76%
IsamMafool	Corpus 4	27048	755	718	37	95.09%
Arabic Jamah	Corpus 4	27048	1203	1117	86	92.85%
All Classes	Corpus 4	27048	15856	14098	1758	88.91%

summarized results of infix stemming and in figure 3 we have presented the visualized results of proposed infix stemming.

4) EVALUATION OF PROPOSED POSTFIX RULE

Processed words after the application of prefix and infix rules are then used for postfix stripping rules. The stemming

TABLE 17. Sumarized stemming results of proposed infix stripping rules.

Stemming Characteristics	Corpus
Words consider infix stemming	27048
Words already free from infix's	11192
Infix to be removed from words	15856
Correctly removed infix's	14098
Proposed Stemming Accuracy Results	
Precision	88.9%
Recall	93.5%
F1-Measure	91.1%

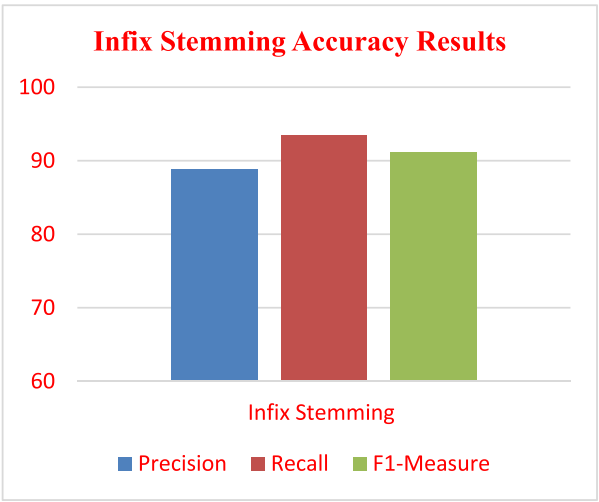

FIGURE 3. Visualized results of infix stemming.

TABLE 18. Stemming accuracy results of proposed postfix rules.

Corpora	Total Words Tested	Number of Words that Matched Postfix Rules	True Positive	False Positive	Accuracy%
Corpora 1	2797	1280	1165	115	91.01%
Corpora 2	1709	960	865	95	90.10%
Corpora 3	6686	4035	3560	475	88.22%
Corpora 4	11192	6275	5590	685	89.08%

TABLE 19. Summarized stemming results of proposed postfix rules.

Stemming Characteristics	Corpus
Words consider postfix stemming	11192
Words already free from suffixes	4917
Suffixes to be removed from words	6275
Correctly removed suffix's	5590
Proposed Stemming Accuracy Results	
Precision	89.0%
Recall	93.8%
F1-Measure	91.3%

accuracy results produced by the proposed generic postfix rules are given in Table-18. In Table 19, we have given summarized results of proposed suffix stripping rules and figure 4 is showing the visualized results.

5) EVALUATION OF PROPOSED ADD CHARACTER LISTS

To normalize the stem as produced after the application of stemming rules, we applied our proposed add characters.

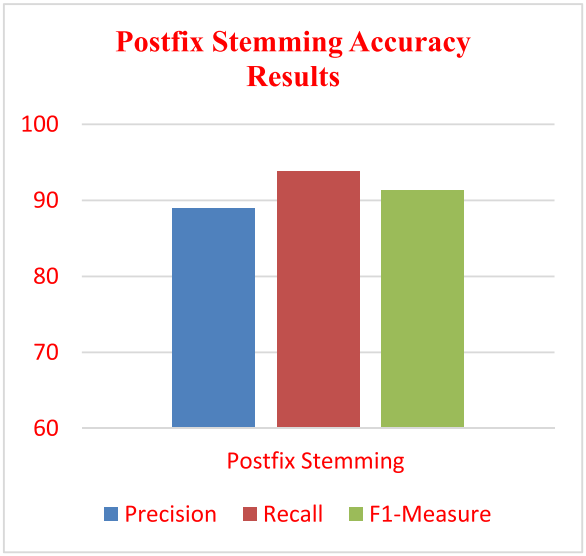

FIGURE 4. Visualized results of postfix stemming.

TABLE 20. Stemming accuracy results of proposed add character lists (ACLS).

Character Name	Number of Words that Matched Proposed Character	True Positive	False Positive	Accuracy %
الف	193	160	33	82.90 %
ت	205	183	22	89.26 %
ر	70	65	5	92.85 %
س	77	67	10	87.01 %
ن	62	53	9	85.48 %
و	36	29	7	80.55 %
ہ	176	141	35	80.11 %
ي	196	165	31	84.18 %
الف،ت،ر،س ن،ي،ہ،و	1015	863	152	85.02 %

TABLE 21. Stemming accuracy results achieved by using prefix rules of light weight urdu stemmer.

Corpora	Total Word Tested	Number of Words that Matched Prefix Rule	True Positive	False Positive	Accuracy %
Corpus 1	4819	920	154	766	16.73 %
Corpus 2	2943	413	57	356	13.80 %
Corpus 3	24238	2238	288	1950	12.86 %
Corpus 4	32000	3571	499	3072	13.97 %

The results obtained by using these characters are presented in Table-20.

C. EXPERIMENT 2: COMPARISON OF PROPOSED APPROACH WITH A LIGHT WEIGHT URDU STEMMER

This experiment is performed to compare the stemming accuracy results of proposed Urdu stemmer with the existing state-of-the art approach i.e. A Light Weight Urdu Stemmer [25]. The evaluation of this experiment also depicts that proposed stemmer is a generic Urdu stemmer and can be applied on any kind of Urdu text dataset. This experiment is conducted on the same Urdu headlines news datasets that are used in experiment 1 and discussed in Table-12.

TABLE 22. Stemming accuracy results achieved by using postfix rules of light weight urdu stemmer.

Corpora	Total Word Tested	Number of Words that Matched Prefix Rule	True Positive	False Positive	Accuracy %
Corpus 1	4819	2760	1520	1240	55.07 %
Corpus 2	2943	1835	840	995	45.77 %
Corpus 3	24238	20023	7990	12033	39.90 %
Corpus 4	32000	24618	10350	14268	42.04 %

TABLE 23. Stemming results achieved by using add character lists (ACLS) of lightweight urdu stemmer.

Character Name	No's of time Character Applied	Correct Applied	False Applied	Accuracy %
الف	280	150	130	53.57%
ت	55	47	8	85.45 %
ن	36	29	7	80.55 %
ہ	318	26	49	84.59 %
ي	48	41	7	85.41 %
ہ،ن،ت،الف،ي	737	536	201	72.72 %

TABLE 24. Comparative results of proposed approach and light weight urdu stemmer.

Approaches	Corpora	Prefix Accuracy	Infix Accuracy All Classes	Postfix Accuracy	ACLs Avg Accuracy
Proposed Approach	C1	85.64%		91.01%	
	C2	87.91%		90.10%	85.02%
	C3	83.59%		88.22%	
	C4	85.28%	88.91%	89.08%	
A Light Weight Stemmer	C1	16.73%		55.07%	
	C2	13.80%		45.775	72.72%
	C3	12.86%		39.90%	
	C4	13.97%	Nil	42.04%	

TABLE 25. Classification accuracies obtained using different configurations of proposed short urdu text classification framework.

Configuration	Classification Accuracy (%)
News classification using Stem less data	82.2
News classification using Prefix and Postfix Stemmed data	83.7%
News classification using Infix, Prefix and Postfix Stemmed data	85.85
News classification using Infix, Prefix and Postfix Stemmed data + conditional probabilities enhancement	87.4%

The application of competitor stemming rules is applied on the 32000 unique words extracted from the experimental datasets. The results produced by the application of competitor rules i.e. prefix rules, postfix rules, and add characters are given in Table-21, Table-22, and Table-23. After the experimental evaluation of exiting approach [25], it is observed that their stemming accuracy is highly affected by the wrong interpretation of prefixes and postfixes. A large numbers of compound words and loan words are also affected due to in-appropriate stemming rules. In order to demonstrate the effectiveness and adoptability of proposed Urdu stemmer, a comparison of proposed approach with A Light Weight Urdu stemmer is given in Table-24.

D. EXPERIMENT 3: EVALUATION OF PROPOSED SHORT URDU TEXT CLASSIFICATION APPROACH

This experiment aims to evaluate the performance of proposed short text classification approach, specifically for Urdu language. The experiment has been conducted on Urdu News Headlines dataset containing headlines from 4 news categories including Politics, Sports, Terrorism and Weather. There are total of 5000 news headlines from different news categories. Labeled training data is extracted by randomly selecting 50% of the news headlines from each news categories as training data. The remaining half of the dataset is then treated as test data. The probabilistic model for each news category using the classified training data is learned using the frame work as presented in section 5 and 6. Test data is then classified using the learned probabilistic model using eq. (5). We computed the classification accuracies using four different configurations of our system to highlight the contributions of different components of the proposed short Urdu text classification framework: i) News classification using stem less data ii) News classification using Prefix and Postfix Stemmed data iii) News classification using Infix, Prefix and Postfix Stemmed data iv) News classification using Infix, Prefix and Postfix Stemmed data + conditional probabilities enhancement. The classification results obtained using the abovementioned five configurations are presented in Table 25. It can be observed from the table that the proposed stemming approach comprising of our novel infix stemming algorithm along with the prefix and postfix stemming rules enhances the classification accuracies of the overall news categorization system. The comprehensive stemming results in generating OR general mature probabilistic model which would otherwise be distorted due to improper stemming of words. It can also be noticed that our proposed conditional probabilities enhancement mechanism further improves the performance of short text classification system.

TABLE 26. URDU text classification accuracies obtained using our proposed framework as compared to the competitors.

Approach	Classification Accuracy (%)
Urdu Text Classification using Naive Bayes [28]	75.3%
Urdu Text Classification using SVM [28]	78.5%
Proposed Comprehensive Urdu Text Classification Approach	87.4%

E. EXPERIMENT 4: COMPARISON OF PROPOSED SHORT TEXT CLASSIFICATION APPROACH WITH COMPETITORS

The purpose of this experiment is to compare the performance of proposed news headline classification approach with the competitors. The proposed system is compared with Naive Bayes and SVM based Urdu text classification approach as presented in [26]. As evident from the results, the proposed comprehensive Urdu text pre-processing and classification system performs significantly better than the approaches presented by the competitor. Ali and Ijaz [26] propose the unsuitability of applying stemming to Urdu language. However, our proposed stemming algorithm which involves enhancement of prefix and postfix stemming whilst proposing novel infix stemming rules, is extremely important and enhances the effectiveness of processing of Urdu text including clustering and classification. These results presented in Table-26 clearly depict the effectiveness of proposed stemming rules on Urdu text classification. Further application of conditional probabilities enhancement and probabilistic classification approach enhances the performance of proposed comprehensive classification approach as compared to the competitors.

VIII. CONCLUSION

In this paper, we present a comprehensive framework to perform short Urdu text classification. One of the major contributions of this research is the demonstration of novel stemming approach that can handle prefix, postfix and infix scenario of stemming. We have presented an effective stemming method for Urdu language that is centered on a rule based affix stripping approach. Due to the robust morphological structure of Urdu, the development of an effective stemmer that has the ability to generate the stem of any kind of Urdu word as well as loan words was a challenging task. To cope with this challenge, we developed different stemming rules i.e. minimum word length rule, prefix, infix, and postfix rules in this proposed Urdu stemmer. These proposed stemming rules are generic and can be applied on any kind of Urdu text. In this stemmer we have introduced novel Urdu infix word classes and infix stripping for these proposed infix classes. These Urdu infix words classes are Alif Arabic Masdar(infinitive verbs beginning with Alif), Te Arabic Masdar (Infinitive verbs beginning with Te), Isam Fiale (Active subject), Isam Mafool (passive object), ArabicJamah(Arabic plural words), and Isam Zarf Makaan (place showing noun). The experimental evaluation of proposed Urdu stemmer delivers remarkable stemming accuracy results on different Urdu textual corpora as compared to the competitor's approach. This Urdu stemmer can be used in Urdu text mining applications, information retrieval system, and natural language processing applications as well.

We have further presented a probabilistic approach for short text classification. The learned probabilities are further enhanced by penalizing the conditional probabilities of words involved in misclassification on news headline to a particular class. It has been shown that this enhancement of conditional probabilities contribute towards improvement in the effectiveness of proposed probabilistic classification approach. Overall, the proposed framework is a comprehensive short Urdu text classification system providing complete solution for pre-processing as well as high level tasks including grouping, classification etc. The proposed probabilistic short text classifier is equally applicable for processing short text in any other language given that we employ appropriate stemmers for the given language.

In future, the proposed work can be extended by analyzing the efficacy of generation of conditional probability model of news classes by employing combination of words using n-gram approach. The proposed set of rules can be improved by Identification of other stemming rules. To further enhance the coverage and efficiency of proposed framework, stemming approach can be studied and revised in future. The approach to enhance conditional probabilities can be revisited to ensure improvement of the quality of probabilistic model for short text classification. Applicability of deep learning approach to text classification can also be considered.

APPENDIX A

INFIX STEMMING RULES

Class	Rule-1	Rule-2	Rule-3	Rule-4	Rule-5
Alif Arabic Masdar	Word start = الف Length = 5 Remove = All الف	Word start = الف Length = 5 Remove = All الف, ت, س, ى, ں, ي, ء, و, ء	Word start = الف Char at index one = ت Length = 5 Remove = All الف,	Word start = الف Char at index two = س Length < 5 Remove = الف, ت,	Word start = الف Char at index three = س Length < 5 Remove = الف, ت,
			ي, ں, ي, ي, و, ء, ء	ي, ں, ي, و, ء, ه, ے	ي, ں, ي, و, ء, ے
Te Arabic Masdar	Word start = ت and contains الف Remove = الف, ت, ں, ي, ے,	Word start = ت Char at length-1 = ي Remove = ت, ں, ے,	-------	-------	-------
IsamFiale	Word Contains الف Length = 4 Remove = All الف	Char at length-1 = ي Length = 4 Remove = ي	-------	-------	-------
IsamMafool	Start = م Char at length-1 = و Length = 5 Remove = م, و	-------	-------	-------	-------
Arabic Jamah	Char at length-1 = و Length = 4 Remove = و	-------	-------	-------	-------
Arabic Jamah and IsamFiale	Word start = م and contains الف Remove = الف, ت, ں, ي, ے, ه	Word start = م Char at index two = ت Length = 5 م, ت, ں, ي			

APPENDIX B

A. PREFIX GLOBAL EXCEPTION LIST SAMPLES

ابواب **(Chapters)**	بادل **(Cloud)**	بلاغت **(Adolescence)**	تاجدار **(Crowned)**
خوشاب **(Khusab)**	خوشامد **(Happy)**	تالاب **(Pool)**	پرستش **(Worship)**
بدعت **(Customs)**	برائے **(For)**	باختہ **(Bakhtiar)**	اہلکار **(Officers)**

B. INFIX GLOBAL EXCEPTION LIST SAMPLES

کسان **(Farmers)**	ماحول **(Environment)**	مالکان **(Owners)**	متاثر **(Affected)**
متوازن **(Balanced)**	تاوان **(Ransomed)**	ترجمان **(Translator)**	اتوار **(Sunday)**
معمار **(Architect)**	اثاثہ **(Asset)**	اداکار **(Actor)**	امارات **(Buildings)**

C. POSTFIX GLOBAL EXCEPTION LIST SAMPLES

مستان **(Busy)**	آستانہ **(Astana)**	شہسوار **(Rider)**	انجانے **(Unknown)**
روزگار **(Employment)**	نسوار **(Naswar)**	نمبردار**(Headman)**	عرصہ **(Time)**
قصبہ **(Town)**	جلوہ **(Manifestation)**	فالتو **(Spare)**	جفاکش **(Hard work)**

APPENDIX C
ADD CHARACTER LIST SAMPLES

Add الف	**Add ت**
کپڑا = الف + کپڑ	ثبت = ت + ثب
کچرا = الف + کچر	آفت = ت + آف
Add ر	**Add س**
ازر = ر + از	حبس = س + حب
مزر = ر + مز	خلس = س + حل
Addن	**Add و**
امن = ن + ام	گفتگو = و + گفتگ
دھن = ن + دھ	چھو = و + چھ
Add ہ	**Add ي**
چڑہ = ہ + چڑ	اسمبلی = ی + اسمبل
بڑہ = ہ + بڑ	الماری = ی + المار

REFERENCES

[1] B. Drury, L. Torgo, and J. J. Almeida, "Classifying news stories to estimate the direction of a stock market index," presented at the 6th Iberian Conf. Inf. Syst. Technol. (CISTI), Jun. 2011, pp. 1–4.

[2] A. Heß, P. Dopichaj, and C. Maaß, "Multi-value classification of very short texts," in *Proc. 31st Annu. German Conf. Adv. Artif. Intell.*, 2008, pp. 70–77.

[3] M. W. Pope, "Automatic classification of online news headlines," M.S. thesis, School Inf. Library Sci., Univ. North Carolina Chapel Hill, Chapel Hill, NC, USA, Nov. 2007.

[4] X. Liu, G. Rujia, and S. Liufu, "Internet news headlines classification method based on the N-Gram language model," in *Proc. Int. Conf. Natural Lang. Process. Knowl. Eng.*, Aug. 2012, pp. 826–828.

[5] R. R. Deshmukh and D. K. Kirange, "Classifying news headlines for providing user centered E-newspaper using SVM," *Int. J. Emerg. Trends Technol. Comput. Sci.*, vol. 2, no. 3, pp. 1–4, May/Jun. 2013.

[6] D. K. Kirange and R. R. Deshmukh, "Emotion classification of news headlines using SVM," *Asian J. Comput. Sci. Inf. Technol.*, vol. 2, no. 5, pp. 104–106, 2012.

[7] I. Dilrukshi, K. De Zoysa, and A. Caldera, "Twitter news classification using SVM," in *Proc. 8th IEEE Int. Conf. Comput. Sci. Educ. (ICCSE)*, Apr. 2013, pp. 287–291.

[8] Q.-U.-A. Akram, A. Naseer, and S. Hussain, "*Assas-Band,* an affix-exception-list based Urdu stemmer," in *Proc. 7th Workshop Asian Lang. Resour.*, Singapore, 2009, pp. 40–47.

[9] M. Al-Khuli, *A Dictionary of Theoretical Linguistics: English-Arabic With an Arabic-English Glossary*. Published by Library of Lebanon, 1991.

[10] S. A. Khan, W. Anwar, and U. I. Bajwa, "Challenges in developing a rule based urdu stemmer," in *Proc. 2nd Workshop South Southeast Asian Natural Lang. Process. (WSSANLP)*, Chiang Mai, Thailand, 2011, pp. 46–51.

[11] C. Bento, A. Cardoso, and G. Dias, "Progress in artificial intelligence," in *Proc. 12th Portuguese Conf. Artif. Intell.*, 2005, pp. 693–701.

[12] M. F. Porter, "An algorithm for suffix stripping," *Program*, vol. 14, no. 3, pp. 130–137, 1980.

[13] J. Mayfield and P. McNamee, "Single N-gram stemming," in *Proc. 26th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2003, pp. 415–416.

[14] M. Melucci and N. Orio, "A novel method for stemmer generation based on Hidden Markov Models," in *Proc. Conf. Inf. Knowl. Manage. (CIKM)*, 2003, pp. 131–138.

[15] M. Bacchin, N. Ferro, and M. Melucci, "Experiments to evaluate a statistical stemming algorithm," in *Proc. Workshop Notes CLEF*, 2002, pp. 161–168.

[16] J. Lovins, "Development of a stemming algorithm," *Mech. Transl. Comput. Linguistics*, vol. 11, nos. 1–2, pp. 22–31, 1968.

[17] D. John, "Suffix removal and word conflation," *ALLC Bull.*, vol. 2, no. 3, pp. 33–46, 1974.

[18] M. F. Porter. (2001). *Snowball: A Language for Stemming Algorithms*. [Online]. Available: http://www.snowball.tartarus.org/texts/introduction.html

[19] D. C. Paice, "Another stemmer," *ACM SIGIR Forum*, vol. 24, no. 3, pp. 56–61, 1990.

[20] S. Khoja and R. Garside, "Stemming Arabic text," Dept. Comput., Lancaster Univ., Lancaster, U.K., Tech. Rep., 1999. [Online]. Available: http://www.scirp.org/(S(lz5mqp453edsnp55rrgjct55))/reference/ReferencesPapers.aspx?ReferenceID=2004190

[21] N. Thabet, "Stemming the Qur'an," in *Proc. Workshop Comput. Approaches Arabic Script-Based Lang.*, 2004, pp. 85–88.

[22] M. Tashakori, M. Meybodi, and F. Oroumchian, "Bon: First persian stemmer," in *Proc. 1st Eurasia Conf. Adv. Inf. Commun. Technol.*, Tehran, Iran 2002, pp. 487–494.

[23] A. Mokhtaripour and S. Jahanpour, "Introduction to a new Farsi stemmer," in *Proc. 15th ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, Arlington, VA, USA, 2006, pp. 826–827.

[24] P. Majumder, M. Mitra, S. K. Parui, G. Kole, P. Mitra, and K. Datta, "YASS: Yet another suffix stripper," *ACM Trans. Inf. Syst.*, vol. 25, no. 4, 2007, Art. no. 18.

[25] S. Ahmad, W. Anwar, U. I. Bajwa, and X. Wang, "A light weight stemmer for Urdu language: A scarce resourced language," in *Proc. 3rd Workshop South Southeast Asian Natural Lang. Process. (SANLP)*, Mumbai, India, 2012, p. 691778.

[26] A. R. Ali and M. Ijaz, "Urdu text classification," in *Proc. 7th Int. Conf. Frontiers Inf. Technol.*, Islamabad, Pakistan, 2009, Art. no. 21.
[27] K . Anita, "TEXT CATEGORIZATION: Building a KNN classifier for the Reuters-21578 collection," Tech. Rep., Dec. 2006. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.135.9946
[28] Y. Jia, Z. Chen, and S. Yu, "Reader emotion classification of news headlines," in *Proc. Int. Conf. Natural Lang. Process. Knowl. Eng. (NLP-KE)*, Sep. 2009, pp. 1–6.
[29] K. Khushbu, "Short text classification using KNN based on distance function," *Int. J. Adv. Res. Comput. Commun. Eng.*, vol. 2, no. 4, pp. 1–4, Apr. 2013.
[30] M. Ali, S. Khalid, M. H. Saleemi, W. Iqbal, A. Ali, and G. Naqvi, "A rule based stemming method for multilingual Urdu text," *Int. J. Comput. Appl.*, vol. 134, no. 8, pp. 10–18, Apr. 2016.
[31] M. Ali, S. Khalid, and M. H. Saleemi, "A novel stemming approach for Urdu language," *J. Appl. Environ. Biol. Sci.*, vol. 4, no. 7S, pp. 436–443, 2014.
[32] S. Khan, W. Anwar, U. Bajwa, and W. Xuan, "Template based affix stemmer for a morphologically rich language," *Int. Arab J. Inf. Technol.*, vol. 12, no. 2, pp. 146–154, 2015.
[33] S. M. J. Rizvi and M. Hussain, "Analysis, design and implementation of Urdu morphological analyzer," in *Proc. Student Conf. Eng. Sci. Technol. (SCONEST)*, Aug. 2005, pp. 1–7.
[34] N. Durrani, "Typology of word and automatic word Segmentation in Urdu text corpus," M.S. thesis, Dept. Comput. Sci., Nat. Univ. Comput. Emerg. Sci., Lahore, Pakistan, 2007.
[35] *Urdu Lughat*. Accessed: Jan. 5, 2017. [Online]. Available: http://urdulughat.info/

MUBASHIR ALI received the M.S. degree from Bahria University, Islamabad, Pakistan, in 2014. He is currently an Assistant Professor with the Department of Computer Science and Information Technology, University of Lahore, Pakistan. He has seven years of software development experience in research and development-based public sector organizations in Pakistan. He has authored over ten publications in international journals and conferences. His active research areas are NLP, Data mining, machine learning, software repository mining, and business intelligence.

SHEHZAD KHALID received the graduation degree from the Ghulam Ishaq Khan Institute of Engineering Sciences and Technology, Pakistan, in 2000, the M.Sc. degree from the National University of Science and Technology, Pakistan, in 2003, and the Ph.D. degree from the University of Manchester, U.K., in 2009. He is a Professor and the Head of the Department of Computer Engineering. He is a qualified academician and a researcher with over 70 International publications in conferences and journals. He has also authored various books and book chapters.

MUHAMMAD HASEEB ASLAM received the bachelor's degree (*cum laude*) in computer engineering from Bahria University, Islamabad. He is the Gold Medalist of the Batch 2013–2017. He was featured in TechJuices' 25 under 25. Each year TechJuice publishes a list of high achieving Pakistanis under the age of 25 (or 25) who have excelled in technology, Haseeb was seventh on the said list. He was also a recipient of the Richard E. Merwin Student Scholarship from the IEEE Computer Society which is a U.S. $1000 Cash Award, and the IEEE CS Ambassador of the Asia–Pacific region. His research interests include, image processing, medical robotics, and brain–computer interface.