An Extended Pattern Based Comprehensive Stemmer for the Urdu Language

MUBASHIR ALI, School of Computer Science, University of Birmingham, Birmingham, United Kingdom of Great Britain and Northern Ireland
ANEES BAQIR, Data Science & AI, Northeastern University London, London, United Kingdom of Great Britain and Northern Ireland and Complex Human Behavior Lab, Fondazione Bruno Kessler, Trento, Italy
HAFIZ HUSNAIN RAZA SHERAZI, School of Computing, Newcastle University, Newcastle upon Tyne, United Kingdom of Great Britain and Northern Ireland
SHEHZAD KHALID, Bahria University Faculty of Engineering Sciences, Islamabad, Pakistan
PHILLIP SMITH, School of Computer Science, University of Birmingham, Birmingham, United Kingdom of Great Britain and Northern Ireland
MARK LEE, School of Computer Science, University of Birmingham, Birmingham, United Kingdom of Great Britain and Northern Ireland

The Urdu language is used by approximately 200 million people for spoken and written communications on a daily basis. There is a substantial amount of unstructured Urdu textual data that is available worldwide. Data mining techniques can be used to extract meaningful knowledge from such a large, potentially informative source of data. There are many text processing systems available to process unstructured textual data. However, these systems are mostly language specific and developed for a variety of languages such as English, Spanish, Chinese, and so on. Unfortunately, there are not as many language processing resources available for Urdu. Stemming is one of the most important preprocessing steps in the text mining process and its goal is to reduce grammatical words form, e.g., parts of speech, gender, tense, and so on, to their root form. In this work, we have extended the stemming capabilities of our existing pattern-based comprehensive stemming system for Urdu text. In addition to the existing stemming rules in previous work, we introduce novel stemming rules for prefix, and infix stemming. We also optimize the existing suffix removal rules and extend the add character lists for word normalization. These stemming rules are generic and have the ability to generate the stem of Urdu words as well as loan words (words belonging to other languages i.e., Arabic, Persian, Turkish). In the experimental evaluation, we have observed a significant improvement in the overall stemming accuracy of our proposed pattern-based Urud stemmer, which demonstrates the adoptability of the proposed stemming approach for a variety of text-processing applications.

Authors' Contact Information: Mubashir Ali (Corresponding author), School of Computer Science, University of Birmingham, Birmingham, United Kingdom of Great Britain and Northern Ireland; e-mail: m.ali.16@bham.ac.uk; Anees Baqir, Data Science & AI, Northeastern University London, London, United Kingdom of Great Britain and Northern Ireland and Complex Human Behavior Lab, Fondazione Bruno Kessler, Trento, Trentino-Alto Adige, Italy; e-mail: anees.baqir@nulondon.ac.uk; Hafiz Husnain Raza Sherazi, School of Computing, Newcastle University, Newcastle upon Tyne, Tyne and Wear, United Kingdom of Great Britain and Northern Ireland; e-mail: hafiz.sherazi@uwl.ac.uk; Shehzad Khalid, Bahria University Faculty of Engineering Sciences, Islamabad, Islamabad, Pakistan; e-mail: shehzad@bahria.edu.pk; Phillip Smith, School of Computer Science, University of Birmingham, Birmingham, United Kingdom of Great Britain and Northern Ireland; e-mail: p.smith.7@bham.ac.uk; Mark Lee, School of Computer Science, University of Birmingham, Birmingham, United Kingdom of Great Britain and Northern Ireland; e-mail: m.g.lee@bham.ac.uk.

ACM 2375-4699/2024/11-ART169
https://doi.org/10.1145/3701231

CCS Concepts: • **Computational linguistic → Natural language processing;**

Additional Key Words and Phrases: Urdu stemming, infix rules, stemming rules, stemming lists

ACM Reference Format:
Mubashir Ali, Anees Baqir, Hafiz Husnain Raza Sherazi, Shehzad Khalid, Phillip Smith, and Mark Lee. 2024. An Extended Pattern Based Comprehensive Stemmer for the Urdu Language. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* 23, 12, Article 169 (November 2024), 18 pages. https://doi.org/10.1145/3701231

1 Introduction

Stemming is an essential pre-processing step in the handling of textual data preceding many tasks in **Natural Language Processing** (**NLP**) such as text mining, and **Information Retrieval** (**IR**). In the latter domain in particular, the primary goal of using a stemming system is to improve the search effectiveness so an information retrieval system can respond to the user query accurately. In linguistic morphology, stemming is a process that aims at producing the stem, or root form of the word by reducing its inflected or derived form. Urdu is the national language of Pakistan and a state language of India. It is an Indo-Aryan language and is written from right to left. Urdu is widely spoken in India, specifically, Indian states such as Uttar Pradesh use Urdu as an official language. According to the 2022 edition of Ethnologue, Urdu is the 10th-most widely spoken language in the world, with 230 million total speakers.

The Urdu vocabulary is interesting as it is composed of many other languages i.e., English, Arabic, Persian, Turkish, Hindi, and so on. The word "Urdu" itself belongs to the Turkish language. All these companion languages have complex morphological structures and linguistic intricacies. Due to the robust morphology of these languages, Urdu is in turn an incredibly rich morphological language. Urdu is robust in both inflectional and derivational morphology [1]. The IR system works on the base or root form of a word rather than its inflected or derived form. So, in order to enhance the performance of the IR system, the development of an Urdu stemmer that has the ability to generate the stem of the morphologically rich language is very important. Stemmer is an algorithm that produces the stem of the word. Urdu stemmer produces the stem of a word by removing prefix, infix, and postfix attached to it. For example, stem of the words خبریں (news), خبروں (news), اخبار (newspaper), اخبارات (newspaper), اخباروں (newspaper) is خبر (news).

In this work, we have extended the stemming capabilities of an existing pattern-based Urdu stemmer [3], by introducing a series of new processing rules. Specifically, we developed novel rules for prefix and infix stemming. As this article will demonstrate, the development of these new rules is able to significantly improve the accuracy of Urdu stemming. Additionally, we have optimized the postfix stemming rules and extended the add character lists, which are used for word normalization. These extended rules are generic but also flexible and have the ability to produce the stem of Urdu words as well as loan words (words belonging to other Arabic, Persian, and Turkish).

The contributions of this work are as follows:

— The introduction of novel prefix rules for Urdu prefix stemming.
— The development of novel infix rules for Urdu infix stemming.
— The optimization of existing postfix stemming rules that achieve better postfix stemming accuracy.
— Extended existing **Add Character Lists** (**ACLs**) that normalize the processed word and produce the stem of Urdu words.

This article is structured as follows: Section 2 presents a brief review of the existing stemming state-of-the-art. The proposed Urdu stemming approach and all the stemming rules are lists are

discussed in Section 3. An experimental evaluation of the proposed stemming system is presented in Section 4. Finally, the concluding remarks are provided in Section 5.

2 Literature Review

The complexities of the Urdu language's grammar, morphological structure, and diverse exceptions create considerable difficulties in the development of an effective Urdu stemmer. Consequently, only a limited number of stemmers have been proposed for the Urdu language. The need to create an effective Urdu stemmer is critical, as it enables the efficient retrieval of information, facilitates text mining, and supports the execution of various NLP tasks.

A stemming system can be based on one of the following three approaches: affix stripping, table lookup, and statistical methods [8]. Affix stripping involves the removal of specific letters at certain indices in a word, while also considering the language's grammatical and morphological structure. A well-known stemmer for the English language was developed by [22]. In this classic table lookup approach, each word and its associated stem is stored in a 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 the statistical approach, based on the size of the corpus, word formation rules are developed. Some methodologies include frequency count, n-gram [18], Hidden Markov Models [19], and link analysis [7]. Until now, lots of stemming methods [9, 16, 17, 21, 23–25] have been proposed for a number of languages, such as English [9, 17, 21, 23], Arabic [16, 25], and Persian [20, 24], and so on.

As far as the Urdu language is concerned, very limited efforts have been reported in the literature, such as the Asass-band Urdu stemmer [1], the lightweight stemmer for Urdu text [15], the novel Urdu stemmer [5], the template-based affix stripping Urdu stemmer [14], the comprehensive Urdu stemmer [4], the pattern-based Urdu stemmer [3], the rule-based inflectional and derivational Urdu stemmer [11], and the rule-based Urdu stemmer [10]. Defining rule priorities, variations, and exceptional cases presents another challenge with the rule-based approach. An Urdu stemmer based on rules was proposed by [1], which used prefix and postfix lists, as well as several exceptional lists. If a query word is not found in the "**Prefix Global Exceptional Lists** (**PrGEL**)", the stemmer finds the affix that is longest and associated with the query word and truncates it in order to obtain the stem. Following that, postfixes are checked against the "**Postfix Global Exceptional Lists** (**PoGEL**)", and if none are found, every potential postfix rule is created and checked against the **Postfix Rule Exceptional List** (**PoREL**). Even after this, if the postfix is not found, the stemmer removes the longest substring. Finally, the stemmer adds appropriate letters using the ACLs.

[13] built a database of root words and their frequencies and proposed a stemmer that checks for appropriate affixes and applies relevant rules to produce a list of possible stems. The stem with the greatest frequency is returned. [15] proposed a stemmer that first searches the stop words list for the query word, and if the search returns no results, it looks into the "**Global Prefix Exceptional List** (**GPEL**)". If the prefix is still unidentified, the relevant rules are used to delete it, and the resulting term is verified against the "**Global Suffix Exceptional List** (**GSEL**)". If the suffix cannot be found, the stemmer applies the appropriate criteria, normalizes the stem using the Add Character List, and returns the stem.

Another Urdu stemmer was proposed by [10] that does not utilize exception lists in their study. Using specified prefix and suffix lists, the stemmer recognizes true affixes (prefix/suffix) and eliminates them to extract the stem if found. [5] presented another stemmer for the Urdu language that first compares the query word to a specified list that contains the stop word. If the word is not found in the list, the stemmer searches the PrGEL for it. If the word cannot be found, it excludes the prefix using a predefined list of prefixes. The resulting word is then looked for in the "PoGEL" and returned as the stem if found. If a suitable suffix is present, the stemmer adds the appropriate

letter to the end of the stem after chopping it up, which is then returned as the stem. This proposed stemmer is identical to the one introduced by [15].

In another study, [11] proposed an Urdu stemmer, which produces the stem of Urdu words by removing prefix and suffix attached to the word. This stemming approach was not able to handle the Urdu infix stemming. Furthermore, [6] proposed another Urdu stemmer to cope with the challenge of Urdu infix stemming. In their stemmer authors introduced a few classes for Urdu infix stemming and developed rules based on the proposed Urdu infix classes. Their work significantly improves the stemming accuracy of their existing Urud stemmer [5].

Template-based stemmers use patterns or templates to identify stripping terminations in inflected words based on variables such as word length and letter position. [2] used this approach to derive the correct stem from words. This approach is commonly used for Arabic, Persian, and Urdu languages [12]. While patterns are effective in identifying stripping letters, each pattern can have variations or exceptions, as shown by [14] with their template "افعال (afaal)" which has variations like "تحائف (tahayef/gifts)" and "حساس (hasas/sensitive)". In another study, [3] developed a pattern-based stemmer for Urdu that handles prefixes and suffixes like the earlier stemmer proposed by [5] and identifies infixes using predefined patterns.

In our earlier study [3], we proposed a pattern-based comprehensive stemmer for the Urdu language, which has the ability to produce the stem of Urdu words as well as loan words. We introduced novel infix rules for Urdu infix stemming. These infix rules were based on novel Urdu infix word 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), Arabic Jamah (Arabic plural words), and Isam Zarf Makaan (place showing noun). In this work, we have extended the stemming capabilities of our existing pattern-based comprehensive stemming system [3] for Urdu text. In addition to the existing stemming rules, we in this work, have introduced novel stemming rules for prefix, and infix stemming. We also optimize the existing suffix removing rules and extended the add character lists for word normalization. These stemming rules are generic and have the ability to generate the stem of Urdu words as well as loan words (words belonging to other language i.e., Arabic, Persian, Turkish).

3 Proposed Urdu Stemming System

The development of an effective Urdu stemmer is a challenging task due to the complex morphological structure of the Urdu language. To address this challenge, we have extended the stemming abilities of the pattern-based Urdu stemmer of [3] by introducing new stemming rules for prefix, infix, and postfix stemming. An architecture of the proposed stemming system is given in Figure 1. We introduced novel stemming rules for prefix and infix stemming, and also optimize the existing postfix stemming rules. A complete overview of the proposed stemming system is depicted in Figure 2.

To develop Urdu stemming system, we have also developed different stemming rules such as (prefix, infix, and postfix), and stemming lists such as PrGEL, **Infix Global Exception List** (**InGEL**), PoGEL, add character lists. These stemming rules are discussed in Section 3.1, and stemming lists are explained in Section 3.4.

The working of the proposed Urdu stemming system is based on four different modules. In the first module, stop words are removed from the data set, and then prefix stripping is performed on a given word. After that, the pre-processed word is passed to the second module to perform infix removal on it. The output of the second module is then passed to the third module for postfix stemming, and in the final module stem of the word is produced. Module 1 of prefix removal is further composed of five different steps. In the very first step, a word is selected from the word data set, and then the selected word is compared with a static list of stop words in step 2. The word

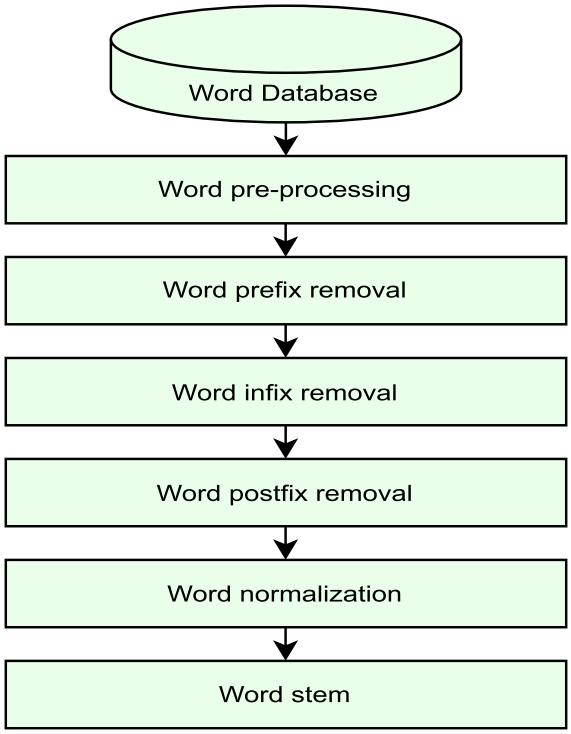

Fig. 1. Architecture of the proposed stemming system.

is filtered out at step 2 if its match is found in the list of less informative words. In step 3, the stem of the word is determined by looking at the word's length, and in step 4, it is checked whether the selected word exists in the PrGEL or not. If the selected word does not exist in PrGEL, then prefix removal rules are applied to it to detach the prefix from it.

The infix stripping module is composed of three different steps. At the first step of the infix removal module, the word is searched in InGEL, if the selected word is found in InGEL, then the rest of the steps of the infix module are ignored and the selected word is passed to module 3 for postfix stemming. But, If the selected word does not exist in InGEL, the infix removing rules are applied to it at step 2. If any one of the infix rules is matched and consequently applied, the processed word is then moved to module 4 in order to generate the stem of the term. On the other hand, if no rule is matched, then the selected word is moved to module 3 for postfix stripping. After the applications of the prefix and infix rules, the word is then passed to module 3 for postfix stemming.

In the third module of the stemming system, postfix stripping is performed. This module is also based on three steps. In step 1, the word is filtered out and marked as a stem word, if it is found in PoGEL. But, if the selected word does not found in PoGEL, then postfix removing rules are applied at step 2, and maximum matched suffixes are removed from the selected word. After the application of postfix rules, the selected word is then passed to module 4 of the system to normalize the processed word and generate the stem word. On the other hand, if no rule match is found then the selected word is considered to be reduced to its stem.

The final module, module 4 is then used to produce the surface form of the selected word if required. After the applications of module 2 and module 3, if the processed word exists in any of the ACLs, then the corresponding character is attached to the end of the word to produce the correct stem. Otherwise, the word is considered as a stem word.

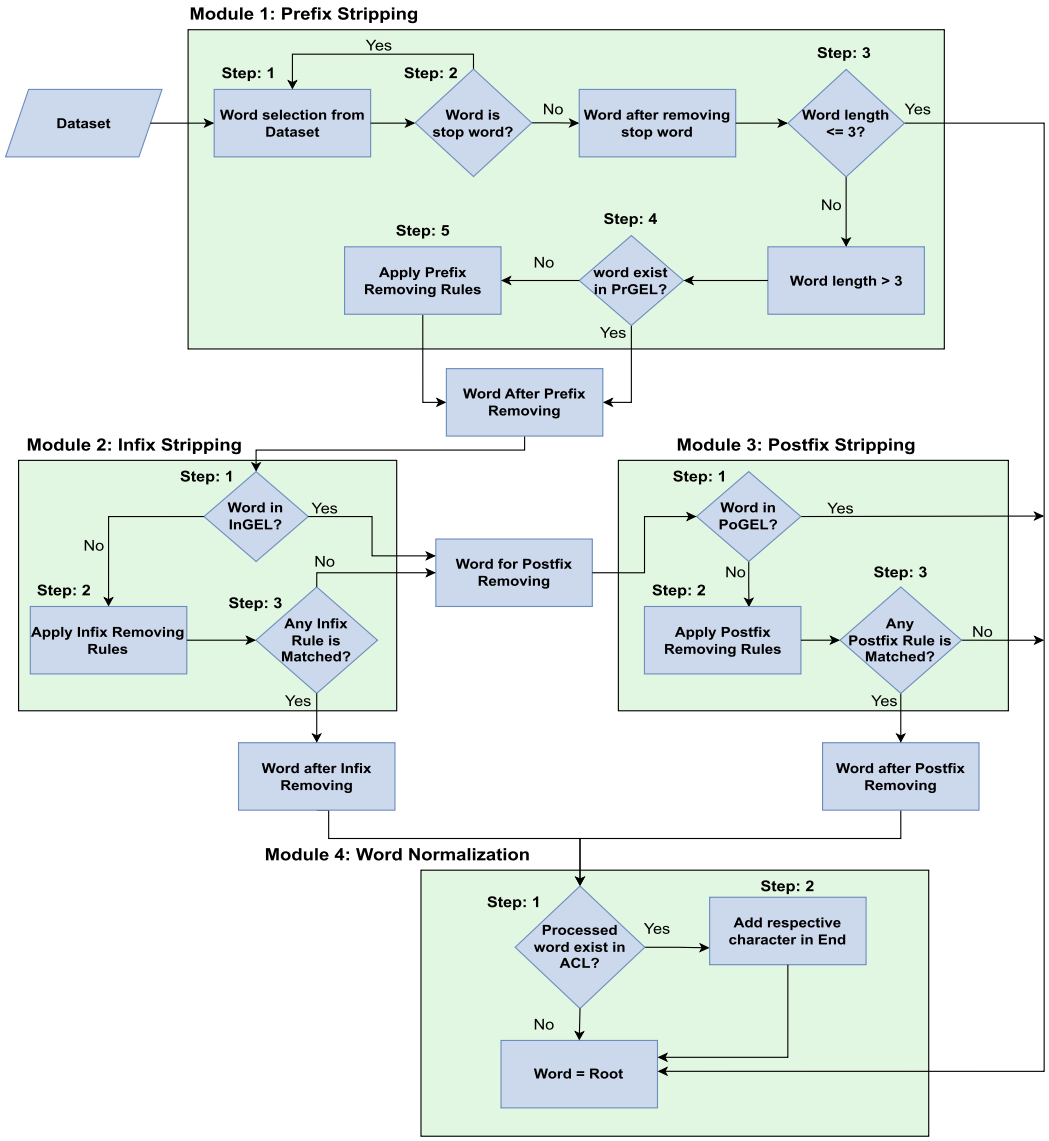

Fig. 2. Flow diagram of the extended Urdu stemming system.

3.1 Stemming Rules

In this work, we have developed three kinds of stemming rules i.e., prefix, infix, and postfix. These rules are generic and can be applied to any kind of Urdu data set to produce the stem. In addition to the rules presented in [3], in this work, we introduce new rules for prefix and infix stemming and also optimize the existing postfix rules.

3.1.1 Prefix Stripping Rules. In Urdu language, a prefix is typically the smallest unit of a word, attached at the beginning of the word. In Urdu morphology, it is known as سابقہ. A prefix may be a combination of one or two characters or maybe a complete word. In this work, we have optimized

Table 1. Examples of Prefix Removing Rules

تا	بد	بے	بر
با	بے	نا	در
ال	یا	لا	از

Table 2. Examples of Words Handled by Proposed Novel Prefix Rules

Proposed Rules	Original Word	Stem Word	Original Word	Stem Word	Original Word	Stem Word
Rule for Tay (ت)	تجمل	جمل	تصور	صور	تشکر	شکر
Rule for Meem (م)	مبشر	بشر	مقرب	قرب	مفکر	فکر
Rule for Mat (مت)	متبرک	برک	متفصل	فصل	متکلم	کلم

existing prefix rules and generated a list of 50 generic prefix rules by reducing existing rules. Some prefix-removal rules are presented in Table 1.

In addition, we also developed new generic rules for prefix stemming to automate the prefix stripping. These rules are as follows:

3.1.2 Rule for Tey. If a word starts with Tey (“ت”), *and the length of the word is exactly equal to four, but not containing Alif* (“الف”),
then remove Tey (“ت”), *at zero indexes from the word.* Examples of words handled by this rule are given in Table 2.

3.1.3 Rule for Meem. If a word starts with Meem (“م”) and the length of the word is exactly equal to four, but not containing alif (“الف”),
then remove Meem (“م”) at zero index from the word. Words handled by this rule are presented in Table 2.

3.1.4 Rule for Mat. If a word starts with Mat (“مت”) and the length of the word is exactly equal to five, but not containing alif (“الف”),
then remove Meem (“م”) and Tey (“ت”) from the beginning of the word. Examples of words handled by this rule are shown in Table 2.

3.2 Infix Stripping Rules

One of the most important contributions of this work is to address the problem of infix stripping by devising generic infix rules. Many parts of Urdu grammar are influenced by Arabic grammar. Therefore, Urdu morphology has inherited features of this parent language. To cope with the challenges of Urdu infix stemming, and to enhance the stemming accuracy of the pattern-based comprehensive stemmer [3], we in this work, with the help of linguistic experts, have developed new generic rules for infix stemming. These rules with the example of words are discussed in this section.

3.2.1 Rule for Alif. If a word starts with Alif (“الف”), and the length of the word is exactly greater than five, and character at index one is Noon (“ن”) and index 2 is != Tey (“ت”),
then remove all the Alif (“الف”), Tey (“ت”), Chhoti-yaee (“ی”), Wao-hamza (“ؤ”), Hamza (“ء”), and Noon (“ن”), from the word. Examples of words handled by this rule are given in Table 4.

3.2.2 Rule for Tey. If a word starts with Tey (“ت”), and the length of the word is exactly greater than five, and the last character of the word is Noon-hay (“نہ”),
then remove all the Alif (“الف”), Tey (“ت”), Noon-hay (“نہ”), ChhotiYeh (“ی”), BadiYeh (“ے”), from the word. Examples of words handled by this rule are shown in Table 4.

3.2.3 Rule for Say. If a word starts with Say (“ث”), and the length of the word is exactly equal to five,
then remove all the Alif (“الف”), Tey (“ت”), Hay (“ہ”), ChhotiYeh (“ی”), BadiYeh (“ے”), Noon (“ن”), Noon-Gunna (“ں”), Wao (“و”), from the word. Examples of words handled by this rule are given in Table 4.

3.2.4 Rule for Hay. If a word starts with Hay (“ح”), and the length of the word is exactly equal to five,
then remove all the Alif (“الف”), Hay (“ہ”), ChhotiYeh (“ی”), ChhotiYeh hamza (“ئ”), BadiYeh (“ے”), Wao (“و”), Hamza (“ء”), from the word. Examples of words handled by this rule are given in Table 4.

3.2.5 Rule for Zal. If a word starts with Zal (“ذ”), and the length of the word is exactly equal to five,
then remove all the Alif (“الف”), ChhotiYeh (“ی”), ChhotiYeh hamza (“ئ”), BadiYeh (“ے”), Hay (“ہ”), from the word. Examples of words handled by this rule are given in Table 4.

3.2.6 Rule for Meem. If a word starts with Meem (“م”), and the last character of the word is also Meem (“م”), and the length of the word is exactly equal to five,
then remove Meem (“م”), at zero index and also remove Wao (“و”), and Tey (“ت”), from the word. Examples of words handled by this rule are given in Table 4.

3.2.7 Rule for Wao. If a word starts with Wao (“و”), and the length of the word is exactly equal to five,
then remove all the Alif (“الف”), Tey (“ت”), Hay (“ہ”), ChhotiYeh (“ی”), ChhotiYeh hamza (“ئ”), BadiYeh (“ے”), from the word. Examples of words handled by this rule are given in Table 4.

3.2.8 Rule for Aeen. If a word starts with Aeen (“ع”), and the length of the word is exactly equal to five,
then remove all the Alif (“الف”), Tey (“ت”), Hay (“ہ”), ChhotiYeh (“ی”), ChhotiYeh hamza (“ئ”), BadiYeh (“ے”), Wao (“و”), from the word. Examples of words handled by this rule are given in Table 4.

3.2.9 Rule for Saad. If a word starts with Saad (“ص”), and the length of the word is exactly equal to five,
then remove all the Alif (“الف”), Tey (“ت”), Hay (“ہ”), ChhotiYeh (“ی”), ChhotiYeh hamza (“ئ”), BadiYeh (“ے”), Wao (“و”), from the word. Examples of words handled by this rule are given in Table 4.

3.2.10 Rule for Zhad. If a word starts with Zhad (“ض”), and the length of the word is exactly equal to five,
then remove all the Alif (“الف”), Tey (“ت”), Hay (“ہ”), ChhotiYeh (“ی”), ChhotiYeh hamza (“ئ”), BadiYeh (“ے”), Wao (“و”), from the word. Examples of words handled by this rule are given in Table 4.

3.2.11 Rule for Sheen. If a word starts with Sheen (”ش“), and the length of the word is exactly equal to five,

then remove all the Alif (“الف”), Tey (“ت”), Hay (“ہ”), ChhotiYeh (“ی”), ChhotiYeh hamza (“ئ”), BadiYeh (“ے”), Noon (“ن”), Noon-Gunna (“ں”), Wao (“و”), from the word. Examples of words handled by this rule are given in Table 4.

3.2.12 Rule for Zuain. If a word starts with Zuain (”ظ“), and the length of the word is exactly greater than five,
then remove all the Alif (“الف”), Tey (“ت”), ChhotiYeh (“ی”), BadiYeh (“ے”), Noon (“ن”), Noon-Gunna (“ں”), Wao (“و”), from the word. Examples of words handled by this rule are given in Table 4.

3.2.13 Rule for Tuain (word_length > 5. If a word starts with Tuein (“ط”), and the length of the word is exactly greater than five,
then remove all the Alif (“الف”), Tey (“ت”), ChhotiYeh (“ی”), ChhotiYeh hamza (“ئ”), BadiYeh (“ے”), Noon (“ن”), Noon-Gunna (“ں”), Wao (“و”), from the word. Examples of words handled by this rule are given in Table 4.

3.2.14 Rule for Tuain. If a word starts with Tuein (“ط”), and the length of the word is exactly equal to five,
then remove all the Alif (“الف”), Tey (“ت”), ChhotiYeh (“ی”), ChhotiYeh hamza (“ئ”), BadiYeh (“ے”), Noon (“ن”), Noon-Gunna (“ں”), Wao (“و”), from the word. Examples of words handled by this rule are given in Table 4.

3.2.15 Rule for Fay. If a word starts with Fay (”ف“), and the length of the word is exactly equal to five,
then remove all the Alif (“الف”), Tey (“ت”), Hay (“ہ”), ChhotiYeh (“ی”), ChhotiYeh hamza (“ئ”), BadiYeh (“ے”), Noon-Gunna (“ں”), Wao (“و”), from the word. Examples of words handled by this rule are given in Table 4.

3.2.16 Rule for Quaaf. If a word starts with Quaaf (”ق“), and the length of the word is exactly equal to five,
then remove all the Alif (“الف”), Tey (“ت”), ChhotiYeh (“ی”), ChhotiYeh hamza (“ئ”), BadiYeh (“ے”), Noon-Gunna (“ں”), Wao (“و”), from the word. Examples of words handled by this rule are given in Table 4.

3.2.17 Rule for Khay. If a word starts with Khay (”خ“), and the length of the word is exactly equal to five,
then remove all the Alif (“الف”), Tey (“ت”), Hay (“ہ”), ChhotiYeh (“ی”), ChhotiYeh hamza (“ئ”), BadiYeh (“ے”), Wao (“و”), from the word. Examples of words handled by this rule are given in Table 4.

3.2.18 Rule for Ray. If a word starts with Ray (”ر“), and the length of the word is exactly equal to five,
then remove all the Alif (“الف”), Tey (“ت”), ChhotiYeh (“ی”), BadiYeh (“ے”), Hamza (“ء”), Noon (“ن”), from the word. Examples of words handled by this rule are given in Table 4.

3.2.19 Rule for Ghaeen. If a word starts with Ghaeen (“غ”), and the length of the word is exactly equal to five,
then remove all the Alif (“الف”), Tey (“ت”), Hay (“ہ”), ChhotiYeh (“ی”), ChhotiYeh hamza (“ئ”), BadiYeh (“ے”), Wao (“و”), Hamza (“ء”), from the word. Examples of words handled by this rule are given in Table 4.

Table 3. Examples of Words Handled by Proposed Infix Post Processing Rule

Original Word	Pre-Stemp	Stem
محسوس	حسس	حس
حقوق	حقق	حق
خطوط	خطط	خط
احباب	حبب	حب
محبوب	حبب	حب

3.2.20 Rule for Wao (word_length == 5). If a word starts with Wao (“و”), and Alif at 2nd index, and the length of the word is exactly equal to five,

then remove all the Alif (“الف”), Tey (“ت”), from the word. Examples of words handled by this rule are given in Table 4.

3.2.21 Rule for Wao (word_length == 4). If a word starts with Wao (“و”), and ChhotiYeh (“ی”), at 2nd index and the length of the word is exactly equal to four,

then remove all the ChhotiYeh (“ی”) from the 2nd index of word. Examples of words handled by this rule are given in Table 4.

3.2.22 Infix Post Processing Rule. If a processed word contains last and the second last character similar,

then remove the last character from the word. Examples of words handled by this rule are given in Table 3.

3.3 Postfix Striping Rules

A postfix is a morpheme that is attached at the end of the word. In Urdu morphology, it is known as لاحقہ. A postfix may consist of one or two characters and sometimes may be a complete word. In our work, we have used a list of 140 suffixes for postfix removal. An example of these suffixes are given in Table 5.

3.4 Stemming Lists

In this Urdu stemming system, we have developed different stemming lists i.e., PreGEL, “InGEL”, PoGEL, stem word dictionary, and ACL.

3.4.1 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 6.

3.4.2 Prefix Global Exceptional List (PreGEL). As the Urdu language is very morphologically rich, it is critical to correctly identify the prefix from Urdu words. The misunderstanding and incorrect processing of prefixes can lead to poor stemming results and loss of useful words, as well. Urdu morphology contains many words that have a prefix attached to them, but often it is not detached because it is assumed to be apart of the word. For example, the word بارش (rain) contains a prefix ب is removed from this word then it produces رش which is incorrect. On the other hand, we cannot remove the prefix ب from the prefix rules list because this prefix generates the stem of many other important words. Therefore, such words should be treated as exceptional cases in order to keep the meanings of words intact. For this purpose, we have developed a list of about 5,000 words. Some sample words from this list are given in the appendix of this article.

Table 4. Example of Words Handled by Proposed Infix Rules

Proposed Rules	Original Words	Stem Word	Original Word	Stem Word	Original Word	Stem Word
Rule for Alif (الف)	انعقاد	عقد	انکشاف	کشف	انفراق	فرق
Rule for Tey (ت)	تعشقانہ	عشق	تعصبانہ	عصب	توصیفانہ	وصف
Rule for Say (ث)	ثقالت	ثقل	ثمارہ	ثمر	ثقابت	ثقب
Rule for Hay(ح)	حاجات	حجت	حافظہ	حفظ	حاکمہ	حکم
Rule for Zal (ذ)	ذبیح	ذبح	ذاکرہ	ذکر	ذواقی	ذوق
Rule for Meem (م)	مرسوم	رسم	محروم	حرم	مقسوم	قسم
Rule for Wao (و)	وزارت	وزر	وصولی	وصل	واسطہ	وسط
Rule for Aeen(ع)	عابدہ	عبد	عرائض	عرض	عاشور	عشر
Rule for Saad (ص)	صابری	صبر	صدارت	صدر	صباحت	صبح
Rule for Zhad (ض)	ضباطہ	ضبط	ضمانت	ضمن	ضربات	ضرب
Rule for Sheen (ش)	شاداب	شدب	شفاعت	شفع	شاعری	شعر
Rule for Zuain (ظ)	ظالمانہ	ظلم	ظالمیت	ظلم	ظریفوں	ظرف
Rule for Tuain(ط) (word_length > 5)	طالبات	طلب	طبقاتی	طبق	طابعات	طبع
Rule for Tuain (ط)	طالبہ	طلب	طماعیت	طمع	طابقہ	طبق
Rule for Fay (ف)	فاجرہ	فجر	فحش	فاحشہ	فاسدہ	فسد
Rule for Quaaf (ق)	قرابت	قرب	قاتلہ	قتل	قاصدہ	قصد
Rule for Khay (خ)	خاتمہ	ختم	خلائق	خلق	خماری	خمر
Rule for Ray (ر)	روابط	ربط	رحمان	رحم	رابعہ	ربع
Rule for Ghaeen (غ)	غارقہ	غرق	غبط	غابطہ	غضارت	غضر
Rule for Wao (و) (word_length == 5)	ولادت	ولد	وکالت	وکل	غضارت	غضر
Rule for Wao (و) (word_length == 4)	وحید	وحد	ولید	ولد	وعید	وعد

Table 5. Examples of Postfix Removing Rules

ویں	یات	ات	وں
یں	جات	ئیں	بات
یز	وے	راں	وار

Table 6. Examples of Stop Words

سے	کا	نے	کی
اس	کے	ان	ہے
وہ	ہیں	ہم	ہم

3.4.3 Infix Global Exceptional List (InGEL). Urdu morphology is influenced by the Arabic grammar so there are many words in Urdu morphology that are from the Arabic 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 the 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 approximately 3,000 words that are known as the infix global exception list. Some sample words from this list are given in the appendix.

Table 7. Examples of Stem Words

ظلم	خبر	قبر	قبر
صدق	صدر	امن	حکم
نشر	قبل	فکر	بشر

3.4.4 Postfix Global Exceptional List (PoGEL). Like 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, there is the possibility that an invalid stem of the word could be 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 given that the removal changes the meaning of the word. In order to maintain the integrity of the meaning of such words, an exception list of approximately 6,000 words has been generated. This list is known as postfix global exception list. Some samples of this exception list are given in the appendix.

3.4.5 Stem Word Dictionary. To check the stemming accuracy of the proposed Urdu stemmer, we have developed a generic stem word dictionary of approximately 15,000 words. 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 such stem words are presented in Table 7.

3.4.6 Add Character Lists (ACLs). In some cases, the execution of proposed infix and postfix rules generates 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 (الف ک، ت، ر، س، ن، و، ہ، ی)

3.5 Proposed Stemming Algorithm

The proposed extended Urdu stemmer algorithm works on the basis of the 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 following steps are applied to generate the stem of Urdu words:

(1) Select a word from a dataset.
(2) Filter out the selected 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 word from the word sequence.
(3) Check the length of the selected word.
    (a) If the length of the word is less than or equal to three then the word is already a stem word.
    (b) If the length of the word is greater than three then go to step 4.
(4) Search for the word in the PreGEL.
    (a) If the word exists in PreGEL then go to step 5.
    (b) If the word does not exist in PreGEL then apply prefix removing rules and remove the maximum matched prefix from the word and go to step 5.
(5) Search for the word in the InGEL.
    (a) If the word is found in InGEL then go step 6.

Table 8. An Overview of Experimental Datasets

Sr.No.	Corpora	Description	Total Words	Unique Words
1	Corpus 1	An Urdu headline news corpus. It contains the news of two different categories: politics and weather	12,500	5,070
2	Corpus 2	Another Urdu headline news corpus. It consists of two different news classes: sports and religion.	7,250	3,080
3	Corpus 3	It consists of unique Urdu words. It has developed by using various grammar books and Urdu dictionaries.	24,238	24,238

Table 9. Accuracy Results of Proposed Prefix Removing Rules

Proposed Rules	Total Words Tested	Words Matched Prefix Rules	True Positive	False Positive	Accuracy %
Rule for Tay (ت)	27,048	574	513	61	89.37%
Rule for Meem (م)	27,048	1,167	1,031	136	88.34%
Rule for Mat (مت)	27,048	217	216	01	99.54%
All prefix rules	27,048	1,958	1,760	198	89.88%

(b) If the word is not found in the InGEL, then apply the infix removing rules.
(c) any one of the infix rules is applied, search the processed word in ACLs.
(d) If the processed word is discovered in any ACLs, then attach the respective character to the end of the processed word. Mark the processed word as stem and go to step 7.
(e) If the processed word does not exist in any ACLs, mark the processed word as stem and go to step 7.
(f) If none of the infix rules is applied, then go to step 6.

(6) Search for the word in the PoGEL List.
(a) If the word is found in PoGEL, mark the processed word as stem and go to step 7.
(b) If the word does not exist in PoGEL, then apply the postfix removing rules.
(c) If any one of the postfix removing rules is matched, then remove the maximum attached suffix from the word and search the processed word in ACLs.
(d) If the processed word is found in any ACLs, then attach the respective character to the end of the processed word. Mark the processed word as stem and go to step 7.
(e) If the processed word is not found in any ACLs, mark the processed word as stem and go to step 7.
(f) If none of the postfix rule is applied then mark the word as stem and go to step 7.

(7) Repeat steps 1–6 for all words in the dataset.

4 Experimental Evaluation

To demonstrate the effectiveness of the proposed extended stemming rules, Urdu raw text from a series of datasets was considered for experimental evaluation. In Section 4.1, we present the summary of the used datasets. In Section 4.2, we present the results achieved by using the extended prefix stemming rules, and in Section 4.3 we present the results of infix stemming rules. The postfix stemming results are given in Section 4.4, in Section 4.6 we presented a comparison of improvements in stemming accuracy with the pattern-based Urdu stemmer from [3].

Table 10. Stemming Accuracy Results of Proposed Infix Rules

Proposed Rules	Total Words Tested	Words Matched Prefix Rules	True Positive	False Positive	Accuracy %
Rule for Alif (الف)	27,048	57	46	11	80.70%
Rule for Tey (ت)	27,048	47	36	11	76.59%
Rule for Say (ث)	27,048	89	61	28	68.53%
Rule for Hay(ح)	27,048	399	363	33	90.98%
Rule for Zal (ذ)	27,048	65	52	13	80%
Rule for Meem (م)	27,048	112	92	20	82.14%
Rule for Wao (و)	27,048	138	130	28	94.20%
Rule for Aeen(ع)	27,048	329	287	42	87.23%
Rule for Saad (ص)	27,048	39	33	06	84.61%
Rule for Zhad (ض)	27,048	56	52	04	92.86%
Rule for Sheen (ش)	27,048	191	158	33	82.72%
Rule for Zuain (ظ)	27,048	19	19	00	100%
Rule for Tuain(ط) (word_length > 5)	27,048	129	113	16	87.60%
Rule for Tuain (ط)	27,048	129	107	22	82.94%
Rule for Fay (ف)	27,048	195	177	18	90.77%
Rule for Quaaf (ق)	27,048	204	188	16	92.16%
Rule for Khay (خ)	27,048	260	224	36	86.15%
Rule for Ray (ر)	27,048	153	137	16	89.54%
Rule for Ghaeen (غ)	27,048	258	241	19	93.41%
Rule for Wao (و) (word_length == 5)	27,048	45	37	8	82.22%
Rule for Wao (و) (word_length == 4)	27,048	45	43	2	95.55%
Infix General Rule	27,048	1,249	1,240	9	99.27%

Table 11. Stemming Accuracy Results of Proposed Postfix Rules

Total Words Tested	Words Matched Postfix Rules	True Positive	False Positive	Accuracy %
11192	1,430	1,295	135	90.55%

4.1 Experimental Dataset

For consistency in reporting our results, We consider the same datasets that were used for the development of pattern-based Urdu stemmer [3]. Experiments are conducted on three corpora of Urdu text, and a brief overview of these Urdu corpora is given as follows in Table 8. We consider Corpus 1 and Corpus 2 because in [3] we want to evaluate the effectiveness of the proposed stemming rules for classification task, while Corpus 3 contains a large amount of unique Urdu words.

4.2 Experimental Evaluation of Proposed Prefix Rules

In order to evaluate the effectiveness of proposed prefix stripping rules, we considered all three corpora and only considered the words obtained after the applications of stop words and minimum word length rules discussed in [3]. To elaborate on the results, we calculate the true positive (correctly stemmed words) and false positive (incorrectly stemmed words) against every

Table 12. Stemming Accuracy Results of Proposed ACLs

Character Name	Words Matched Proposed Characters	True Positive	False Positive	Accuracy %
الف	243	211	32	86.83%
ت	321	293	28	91.27%
ر	111	89	22	80.18%
س	97	77	20	79.38%
ن	92	79	13	85.86%
و	63	55	8	87.30%
ہ	221	203	18	91.85%
ی	307	279	28	90.87%
و،ن الف،ی،ہ، س،ر،ت،	1,455	1,286	169	88.38%

Table 13. Stemming Accuracy Comparison

Approach	Words Tested	Words Matched Rules	True Positive	False Positive	Accuracy %
Prefix Stemming					
Pattern-based	27,048	700	597	103	85.28%
Pattern-based(extended)	27,048	1,958	1,760	198	89.88%
Total accuracy	27,048	2,658	2,357	301	88.67%
Infix Stemming					
Pattern-based	27,048	15,856	14,098	1,758	88.91%
Pattern-based(extended)	27,048	4,208	3,836	391	91.15
Total accuracy	27,048	20,064	17,934	2149	89.38%
Postfix Stemming					
Pattern-based	11,192	6,275	5,590	685	89.08%
Pattern-based(extended)	11,192	1,430	1,295	135	90.55%
Total accuracy	11,192	7,705	6,885	820	89.38
ACLs					
Pattern-based	27,048	1,015	863	152	85.02%
Pattern-based(extended)	27,048	1,455	1,286	169	88.38%
Total accuracy	27,048	2,470	2,149	321	87.00%

stemming rule. The stemming accuracy of the proposed Urdu stemmer is calculated as the ratio of true positives and the number of words that matched stemming rules. The obtained results are presented in Table 9.

4.3 Experimental Evaluation of Proposed Infix Rules

After the applications of prefix stripping rules, we used each processed word for the evaluation of the proposed infix-removing rules. Stemming accuracy results of the proposed infix stripping rules in given in Table 10. The results produced by the applications of infix rules demonstrate the effectiveness of the proposed infix rules.

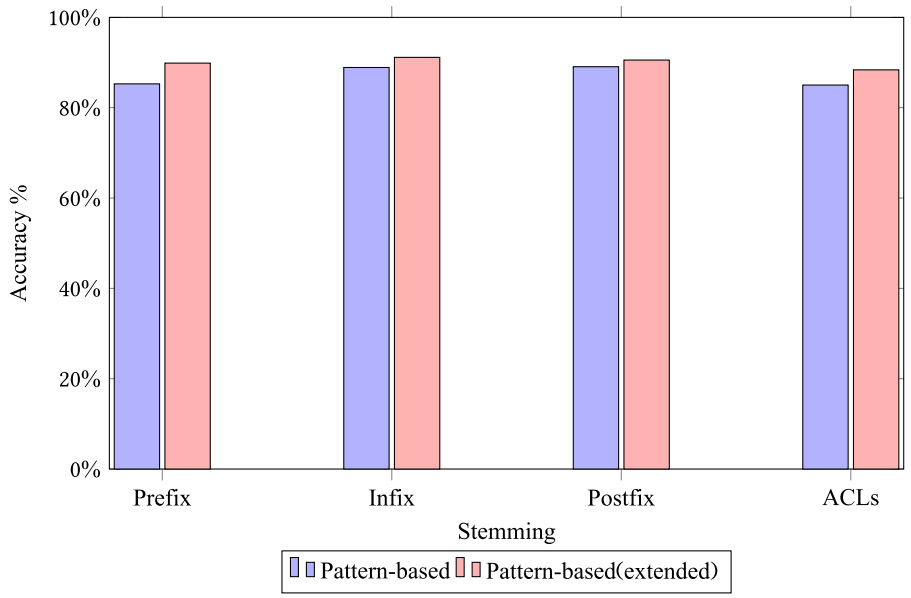

Fig. 3. Accuracy by stemming and approach.

4.4 Experimental Evaluation of Proposed Postfix Rules

After the applications of prefix and infix rules, processed words are then used for postfix stemming. The results achieved by applying the postfix stemming results are given in Table 11. It is obvious from the results, the optimization of existing postfix stripping rules has significantly improved the postfix stemming accuracy.

4.5 Evaluation of Proposed Add Character Lists (ACLs)

In order to normalize and produce a valid stem as produced after the application of prefix, infix, and postfix stemming rules, we applied the applications of our proposed add characters. The results obtained by using these characters are given in Table 12.

4.6 Comparison of Improvement in Stemming Accuracy

In this section, we present a comparison of Urdu stemming accuracy with the existing pattern-based stemming system [3] and with this extended stemming approach. We observed an improvement in accuracy (as shown in Tables 9–12) for all the stemming rules i.e., prefix, infix, suffix, and ACLs. Meanwhile, Table 13 shows the comparison of accuracy of aforementioned approaches with existing pattern-based stemming system [3] and extended pattern-based stemming system. Figure 3 presents the visualized results. It is obvious from the results, the development of new stemming rules has significantly improved the overall stemming accuracy of our proposed Urdu stemmer. We can further improve the accuracy of the proposed stemming rules by adding the words that produce false positives to the respective global exception lists. However, we want to observe how many words are handled and destroyed by each rule.

5 Conclusion

In this article, we present an extended stemming system for Urdu language that is centered on a rule-based affix stripping approach. One of the major contributions of this work is the

demonstration of novel stemming rules to enhance the stemming capabilities of a pattern-based comprehensive Urdu stemmer. Due to the robust morphological structure of the Urdu language, the development of an effective Urdu stemmer that has an 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, in this work we introduced novel stemming rules for prefix and infix stemming. We have also optimized the suffix stripping rules and extended the add character lists to produce the root form of a word. These stemming rules are generic and have the ability to generate the stem of Urdu words as well as loan words (words belonging to borrowed language i.e., Arabic, Persian, Turkish). The experimental evaluation of the proposed stemming rules produces remarkable results and significantly improves the overall stemming accuracy of the proposed rule-based Urdu stemmer. The applications of this proposed Urdu stemmer can be utilized in a variety of Urdu text mining applications, information retrieval system, and natural language processing applications as well. The future objective is to develop novel stemming rules for prefix, infix, and postfix stemming in order to enhance the overall accuracy of Urdu stemming. In addition, we are interested in exploring the use of deep learning models to enhance the stemmer's ability to learn from large datasets.

References

[1] Qurat-ul-Ain Akram, Asma Naseer, and Sarmad Hussain. 2009. Assas-Band, an affix-exception-list based Urdu stemmer. In *Proceedings of the 7th Workshop on Asian Language Resources*. 40–47.

[2] Mohammed N. Al-Kabi, Saif A. Kazakzeh, Belal M. Abu Ata, Saif A. Al-Rababah, and Izzat M. Alsmadi. 2015. A novel root based Arabic stemmer. *Journal of King Saud University-Computer and Information Sciences* 27, 2 (2015), 94–103.

[3] Mubashir Ali, Shehzad Khalid, and Muhammad Haseeb Aslam. 2017. Pattern based comprehensive Urdu stemmer and short text classification. *IEEE Access* 6 (2017), 7374–7389.

[4] Mubashir Ali, Shehzad Khalid, and Muhammad Saleemi. 2019. Comprehensive stemmer for morphologically rich Urdu language. *International Arab Journal of Information Technology* 16, 1 (2019), 138–147.

[5] Mubashir Ali, Shehzad Khalid, and Muhammad Haneef Saleemi. 2014. A novel stemming approach for Urdu language. *Journal of Applied Environmental and Biological Sciences* 4, 7S (2014), 436–443.

[6] Mubashir Ali, Shehzad Khalid, M. Haneef Saleemi, Waheed Iqbal, Armughan Ali, and Ghayur Naqvi. 2016. A rule based stemming method for multilingual Urdu text. *International Journal of Computer Applications* 134, 8 (2016), 10–18.

[7] Michela Bacchin, Nicola Ferro, and Massimo Melucci. 2002. University of Padua at CLEF 2002: Experiments to evaluate a statistical stemming algorithm. In *Proceedings of the CLEF (Working Notes)*.

[8] Carlos Bento, Amílcar Cardoso, and Gaël Dias. 2005. *Progress in Artificial Intelligence: 12th Portuguese Conference on Artificial Intelligence, EPIA 2005, Covilha, Portugal, December 5–8, 2005, Proceedings*. Vol. 3808. Springer.

[9] John Dawson. 1974. Suffix removal and word conflation. *ALLC Bulletin* 2, 3 (1974), 33–46.

[10] Vaishali Gupta, Nisheeth Joshi, and Iti Mathur. 2013. Rule based stemmer in Urdu. In *Proceedings of the 2013 4th International Conference on Computer and Communication Technology*. IEEE, 129–132.

[11] Vaishali Gupta, Nisheeth Joshi, and Iti Mathur. 2015. Design and development of rule based inflectional and derivational Urdu stemmer "Usal". In *Proceedings of the 2015 International Conference on Futuristic Trends on Computational Analysis and Knowledge Management* . IEEE, 7–12.

[12] Abdul Jabbar, Sajid Iqbal, Adnan Akhunzada, and Qaisar Abbas. 2018. An improved Urdu stemming algorithm for text mining based on multi-step hybrid approach. *Journal of Experimental and Theoretical Artificial Intelligence* 30, 5 (2018), 703–723.

[13] Rohit Kansal, Vishal Goyal, and Gurpreet Singh Lehal. 2012. Rule based Urdu stemmer. In *Proceedings of the COLING 2012: Demonstration Papers*. 267–276.

[14] Sajjad Khan, Waqas Anwar, Usama Bajwa, and Xuan Wang. 2015. Template based affix stemmer for a morphologically rich language. *International Arab Journal of Information Technology* 12, 2 (2015).

[15] Sajjad Ahmad Khan, Waqas Anwar, Usama Ijaz Bajwa, and Xuan Wang. 2012. A light weight stemmer for Urdu language: A scarce resourced language. In *Proceedings of the 3rd Workshop on South and Southeast Asian Natural Language Processing*. 69–78.

[16] Shereen Khoja and Roger Garside. 1999. Stemming arabic text. *Lancaster, UK, Computing Department, Lancaster University* (1999).

[17] Julie Beth Lovins. 1968. Development of a stemming algorithm. *Mechanical Translation and Computational Linguistics* 11, 1-2 (1968), 22–31.

[18] James Mayfield and Paul McNamee. 2003. Single n-gram stemming. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*. 415–416.
[19] Massimo Melucci and Nicola Orio. 2003. A novel method for stemmer generation based on hidden Markov models. In *Proceedings of the 12th International Conference on Information and Knowledge Management*. 131–138.
[20] Alireza Mokhtaripour and Saber Jahanpour. 2006. Introduction to a new Farsi stemmer. In *Proceedings of the 15th ACM International Conference on Information and Knowledge management*. 826–827.
[21] Chris D. Paice. 1990. Another stemmer. In *Proceedings of the ACM Sigir Forum*. ACM New York, NY, USA, 56–61.
[22] Martin F. Porter. 1980. An algorithm for suffix stripping. *Program* 14, 3 (1980), 130–137.
[23] Martin F. Porter. 2001. Snowball: A language for stemming algorithms. http://snowball.tartarus.org/texts/introduction.html
[24] Masoud Tashakori, Mohammadreza Meybodi, and Farhad Oroumchian. 2002. Bon: The persian stemmer. In *EurAsia-ICT 2002: Information and Communication Technology: First EurAsian Conference Shiraz, Iran, October 29–31, 2002 Proceedings*. Springer, 487–494.
[25] Naglaa Thabet. 2004. Stemming the Qur'an. In *Proceedings of the Workshop on Computational Approaches to Arabic Script-based Languages*. 85–88.

Received 2 July 2023; revised 8 June 2024; accepted 5 October 2024