Received September 30, 2020, accepted October 12, 2020, date of publication October 15, 2020, date of current version October 28, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3031393

RUTUT: Roman Urdu to Urdu Translator Based on Character Substitution Rules and Unicode Mapping

MOBEEN SHAHROZ[1], MUHAMMAD FAHEEM MUSHTAQ[2], ARIF MEHMOOD[3], SALEEM ULLAH[1], AND GYU SANG CHOI[4], (Member, IEEE)

[1]Department of Computer Science, Khwaja Fareed University of Engineering and Information Technology, Punjab 64200, Pakistan
[2]Department of Information Technology, Khwaja Fareed University of Engineering and Information Technology, Punjab 64200, Pakistan
[3]Department of Information Technology, The Islamia University of Bahawalpur, Punjab 63100, Pakistan
[4]Department of Information and Communication Engineering, Yeungnam University, Gyeongsan 38542, South Korea

Corresponding authors: Gyu Sang Choi (castchoi@ynu.ac.kr) and Arif Mehmood (arifnhmp@gmail.com)

This work was supported in part by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2019R1A2C1006159) and in part by the Ministry of Science and ICT (MSIT), South Korea, under the Information Technology Research Center (ITRC) support program (IITP-2020-2016-0-00313) supervised by the Institute for Information & communications Technology Promotion (IITP).

ABSTRACT Urdu language written in English alphabets for communication is known as Roman Urdu. In pronunciation, both are the same but different in spelling and have different shapes of the alphabet. A survey acknowledges that 300 million people are speaking Urdu and about 11 million speakers in Pakistan from which maximum users prefer Roman Urdu for the textual communication. Today most of the modern technologies like computers and mobile phones using English script, due to this local Urdu user has to use English letters to type Urdu script that is Roman Urdu. In this research, Roman Urdu to Urdu Translator (RUTUT) is proposed that consists of preprocessing methods, rule-based character substitution and Unicode based character mapping techniques. It can transliterate the messages or descriptions from the Roman Urdu script to Urdu script which may help the Urdu speaker to elaborate their message in efficient manners. The focus of this research is to analyze the issues related to the Roman Urdu script to Urdu script transliteration and develop a translator based on the concepts of transliteration. This research analyzed Roman Urdu data and identified different rules-based character substitution techniques that transform the Roman Urdu into Urdu script at fundamental levels. This research is carried out using a python programming language in programming tool Anaconda in Jupiter notebook and user-friendly Graphical User Interface (GUI) created by using Tkinter library. To evaluate the RUTUT, different translational tests are performed and compare those results with famous Google online translator and ijunoon online transliteration. The analyses of results show that the proposed RUTUT approach translates accurately than Google online translator and ijunoon online transliteration.

INDEX TERMS Roman Urdu, transliteration, rule-based approach, character substitution, unicode mapping.

I. INTRODUCTION

The multi-linguistic content rapidly growing on the internet in the last decade. The information retrieval process based on cross-lingual [1] and monolingual gain a lot of attention from the Natural Language Processing (NLP) researcher community World Wide Web (WWW). It was the web of the English language and then become a huge collection of multi-linguistics. When the information retrieval process concentrated on the queries and accessed information in the same language is known as monolingual and cross-lingual focused to access information in several different languages [2]. Some Indo-Aryan languages gain attention from researchers in recent years and Urdu started to get focus because on the web Urdu is a major part of the Asian languages [3].

The researchers of the NLP attract to those languages that have script writing styles from right to left like Urdu and Arabic. Urdu is the national language of Pakistan. There

The associate editor coordinating the review of this manuscript and approving it for publication was Yilun Shang.

are almost 11 million Urdu speakers in Pakistan and more than 300 million in the world [4]. Majority of Urdu speakers present in Pakistan, India, UK, Canada and USA. The foundation of Urdu is lies in Arabic, Persian and most of the South Asian languages. Especially Arabic has been studied deeply and it's also one of the Semitic languages. Punjabi, Pashto, Dari and Farsi (Persian) also follow the right to the left script writing style. These languages belong to Proto Indo-Iranian languages [5] that are widely spoken in regions of South Asia. Free word order characteristic and lack of capital and small words are some terms of similarities in these languages. So, many similarities are found in the writing and speaking styles of these languages but individually each language has its grammar and semantics [6]. Due to this, each language needs separate attention. The syntax and morphological structure of Urdu is consisting of Persian, Sanskrit, English, Turkish and Arabic [7]. That's why its structure is complex than other languages. Urdu language processing in the reserved state because of the availability of fewer linguistic resources and gets less attention from the language engineering community.

The digital electronic media and the internet are growing very rapidly [8]. Data mining [9] and Information retrieval [10] processes are engaged to analyze and maintain the vast amount of data. In this age of technology, all the languages are fully functional in computer devices but almost every machine or computer device using English language standards to write or type messages. The local users of mobile devices are largely using Roman Urdu [11]. When a local user who does not know any basics of English language or English alphabet then how the user understands the message or any social media data that is in Roman Urdu as discussed in the section II. Communication provides the link to every process and leads the way to success. The Roman Urdu and English make computer devices and digital worlds very difficult. When the user uses those devices in the Urdu language then everything will get easy to understand for Urdu speaker.

Data mining tasks and information retrieval that mostly consists of topic modeling, event extraction, decision making, relationship exploration, sentiment analysis gives a deep analysis of NLP. There are some important techniques like stopwords [12] removal, name entity recognition [6], shallow parsing, tokenization, POS tagging [13] and morphological analysis [14] are the major parts of the NLP system. English NLP systems are mature enough but Urdu and Roman Urdu NLP need more concentration [7], [15], [16]. Many researchers wrote survey papers on Urdu and its related issues. Many others perform different tasks like stopwords identification, stemming [17], concept searching and NER but not try to give any rule or standard that translates the Roman Urdu into Urdu script [4], [18].

Roman Urdu is the non-standard language that has not any type of grammar or standards of spelling for the written script. Roman Urdu to Urdu script translation has been done in this research. The Urdu speaker who does not know how to type or write in the English language and what is the meaning of English letters than it is very difficult for the user to understand Roman Urdu. The RUTUT translator is based on three major modules that preprocessing, rules-based character substitution and Unicode based character mapping. These modules are further divided into sub-components or modules that work together and perform the specific operations to translate the Roman Urdu query. The proposed architecture and the rule-based character substitution that consists of 12 rules are the novelty of the proposed research. When a Roman Urdu query gives to the RUTUT translator as input then this query pass through these three modules. Each module applies its functionality and passes it to the next module. In the end, local users can get a proper Urdu script as shown in Figures 1 and 3.

The contribution of this paper is as follows.

- The translation of the Roman Urdu script to Urdu script by Using RUTUT translator.
- The preprocessing filters unnecessary data to make it more effective for further processing.
- The rule-based character substitution depends on rules that convert the Roman Urdu script into a specific form of Roman Urdu.
- The Unicode based character mapping convert the Roman Urdu into Urdu characters by following the design scheme of mapping based on Unicode.
- The performance analyses and comparative analyses evaluated the performance of the RUTUT translator.

The structure of the paper categorised into the following sections: section II presents the Roman Urdu in which the impact and influences of the Roman Urdu language is discussed. Section III explains the related work regarding the efforts of previous researcher and studies that are relevant to the proposed approach. Section IV presents the material and methods of this research. Section V explains the results and discussion based on the proposed methodology. Section VI proposed the comparison of the existing and proposed research. Section VII presents the conclusion of this research.

II. ROMAN URDU

Urdu and English languages are widely used in South Asia for textual communication. Digital media such as WhatsApp, Facebook and SMS [11], etc. are largely used by Roman Urdu users for communication with no standard spelling rules. Roman Urdu developed like ''necessity is the mother of invention'' and also by the great impact of the electronic and digital media. Practically, all the official documents and Government departments are all drafted documents in the English language [15], [19]. It is crowned as the global business language. The English language contains such supreme importance in the global age. Consequently, it is a great need to translate the Roman Urdu language into English. Urdu is the supreme language in Asia for composing research, literature and poetry [4]. Poets manipulated the meaning and politeness of Urdu words in multi-levels from centuries to create memorable verses in beautiful manners. So related facts illustrate the influence of Urdu to English Transliteration.

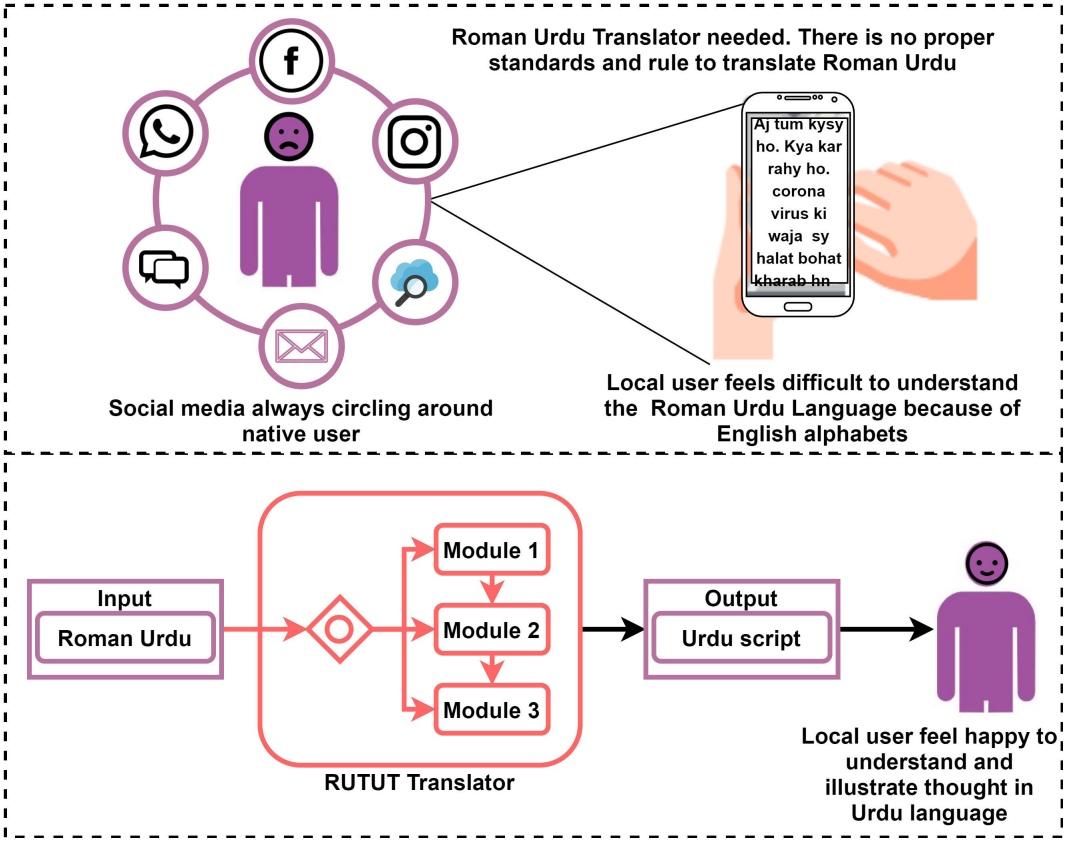
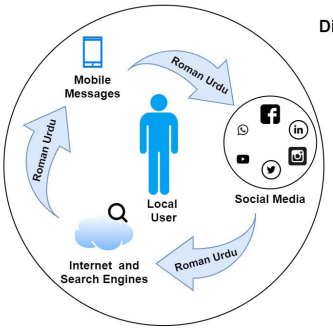


FIGURE 1. The Graphical Abstract of the proposed RUTUT Translator.



Difficulties faced by Users

- Electronic Media mostly used in Roman Urdu in Urdu Speaking Countries Like Pakistan.
- Mostly native users did not know how to spell English letters to create words or sentence to elaborate thoughts clearly in Roman Urdu.
- Native user needs to use Roman Urdu for using Mobile chatting, Social Media, Post Comments and Internet Search Engines etc which is difficult to understand for user.

FIGURE 2. Local users need to use Roman Urdu in daily life.

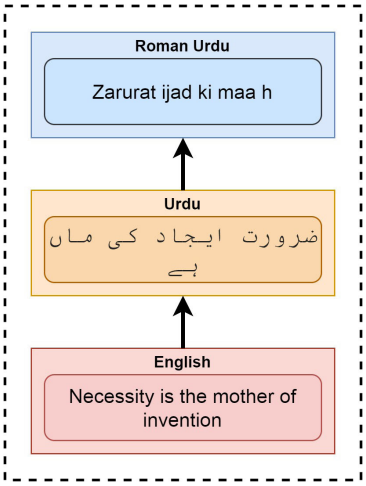


FIGURE 3. A sentence in the form of Roman Urdu, Urdu, and English Languages.

The Pakistani people prefer the Urdu language in the form of Roman Urdu to write poetry, literature and verses. A survey performed in [20] was to illustrate the facts that 80% of Pakistani people use Roman Urdu.

When the user feels uncomfortable in using their mother language then they try to use some English letters for the communication like typing message on SMS or WhatsApp, writing comments on Facebook posts and in the reviews of products etc to elaborate the thoughts in mother language as shown in Figure 2. When writing Urdu language using the Roman script (English letters) then it is known as Roman Urdu. An example is given in Figure 3.

Mobile phone communication was largely used in South Asia by local users, to organize events, maintain social relationships and expressing emotions. Roman Urdu is widely used in text messages for communication. The research is carried out to analyze the structural and linguistic properties of various communication media [21]. Most of the research that relates to mobile data has been carried out on data from developed countries. For text predictions, spelling analyses and spam identification, etc. tasks of linguistic research that was targeting text message written in European languages.

III. RELATED WORK

Many researchers and studies belong to Urdu transliteration [22], [23], Urdu stemming method [20], [24], [25], POS tagging methods [26], sentiment analysis [27], lexical [28] and morphological analysis that gives information of linguistic translation and evolves the Urdu language, are all reviewed in this study. NLP frameworks mature enough for the English language [22] but Urdu NLP framework needs a lot of effort and research to make it mature. Data mining tasks and computational linguistics that depend on the morphological analysis, information extraction, sentiment analysis, part of speech tagging and topic modeling are the methods of NLP. The NLP in the speech and language processing domain [29] contributes to the cognitive modeling, machine translation learning phrases, tera-scale language models, neural networks multi-task, incremental processing and language resource retrieval are the critical significant.

Transliteration algorithms [43], [44] and word dictionaries [45], [46] are used to convert the Roman form of Urdu to Urdu script but accuracy is greatly reduced because of the presence of English words with Roman Urdu in conversational use. Soundex algorithm [47] is used to solve conversational issues that assigns the code to English words on the bases of the position that are then mapped to Urdu script [48]. A lot of efforts and attempts are made in the manual analysis of words for the accent localization. 1736 distinct words are translated into English successfully out of 2000 which shows the accuracy is 86% [49]. Neural machine translation model is based on encoder-decoder architecture [32] using sequence to sequence learning methods. This model consists of two parts one takes the input sentence and the second is responsible for the output. This input is in Roman Urdu form. After getting the input sentence distributes the representation of the source language based on the encoding-decoding architecture that starts the network and riches to the learning dependencies. Then the neural machine translation model translates the Roman Urdu to Urdu script. BLEU [50] evaluation metric is used that is 48.6 on the test.

Roman Urdu to Urdu translation based on the word list [23] gives the novel transliteration approach to the Persio-Arabic letters that have the same in sound but different in written form in Urdu script. The rule-based system used consists of uni-code mapping. This study also identifies the issues involved in translation and different ways of Roman Urdu to Urdu translation. One of the complex issues that handled was one to one mapping is not enough. Emerging new machine and language translation techniques that contribute to a large scale provides the Neural Language Processing and computational linguistics to solve communication problems [26]. The presented approach was consists of three stages. Knowledge-based corpus with its tag set used for tokenization, grammatical rule-based Urdu POS-tagger that presents the syntactical structure and prepared the grammatical structure of sentences for Roman Urdu to English translation. As compared to the Google translator proposed approach in this study gives better results.

The neural network based on the sequence to sequence network model [32], [36], [51] has become a very successful and popular technique to predict the identical sequence for mapping purposes. The kind of problems like handwriting generation, conversational modeling, the secondary structure of protein prediction, question answering, text to speech, music, [52] modeling of polyphonic music, speech recognition, machine translation and modeling of the speech signals are solved by applying these neural network-based models. Further, these concepts of sequence to sequence model based on neural networks are comparatively new but also need relative enhancements. The bidirectional models of [35] include encoder and decoder techniques that take credit for both right to left and left to right orders of sequences. Some improvements in encoder-decoder based modeling but still get only right to left manners [53]. The letter sequence-based character modeling [54] feed the encoders that will create sentences, not with the words but with a sequence of letters. Attention-based models [55] distinguish themselves by drawing toward the center with the appropriate character of the input to the decoder by applying the soft concept of attention.

Transliteration is the subcategory of linguistic translation such that changes of the alphabetic letters of one language into another language is done in transliteration based on similarity measures of the soundings of characters of the target language. Especially, sequence to sequence character mapping technique is mostly implemented in the tasks of the machine learning-based transliteration. The transliteration method has been proposed for linguistics [56] of Sanskrit that translates it into English. Furthermore, the attention-based machine learning approach is utilized to translates the English language into Persian. The most important and challenging part of linguistic engineering and transliteration operations is to identify the semantics, syntax and morphology of the target and source languages. There can be problems of misalignment among the sentences of the source language and the sentences of the target language as well as the length of both languages. In other words, both languages are completely different in every aspect.

Urdu is a morphologically deep complex language but also has some deficiencies and low resources as mentioned by [25]. NLP of the Urdu language statistical method is mention in [57]. Convolutional Neural Network (CNN) based sequence to sequence character mapping techniques, phrased based statistical models of machine learning and conventional techniques based on NLP techniques are used for linguistic translational purposes. Some of fundamental concepts of the translational techniques are based on POS tagging [25], [33], stemming [20], [25], tokenization, lemmatization, annotation [58], named entity recognition and sentence boundary detection [25] are implemented in the different fields. These fields are sentiment analysis, opinion mining, handwriting recognition and detection of plagiarism. The work accomplished for the Urdu language has massively depended on conventional NLP based language handling methods [58], [59] and the utilization of deep learning methods to figuring

TABLE 1. Comparison of the existing systems.

Literature	Summary	Findings	Constraints
[26]	Develop a new translator Transtech that translate Roman Urdu to English suing POS tagger.	Gives better accurate results than the other related work.	Does not handle complex grammatical rule and different variations of Urdu/English sentences in the dataset.
[24]	Hybrid stemming method using Stem dictionary.	Not used any exceptional rule list and word list.	A large storage space is required.
		Good in handling prefixes and suffixes by removing them.	Could not handle infixes.
[30]	Developing a lexical analysis based grammar for Urdu using Hindi wordnet.	Provide the lexical based translator for the Hindi language that helps in Urdu language engineering.	But has some loose ends in complex sentence formations.
[31]	Design and developed rule-based Urdu stemmer USAL using Prefix and suffix list.	Extract the stemming words.	Slow execution because every word passes through the database.
		Retrieve information by removing suffixes and prefixes.	Produce incorrect sentences linguistically.
[32]	Neural Network based Roman Urdu to Urdu Translator proposed with decoding and encoding techniques.	Easy to use and provide fast execution.	Limited length of sentences is allowed only and the results are 48% only.
[33]	Rule-based approach used by using suffixes and prefixes lists.	Fewer lists are used that give fast execution.	The small dataset is used without handling infixes.
		Implementation is easy.	Infixes are not dealing properly.
[34]	Develop an Urdu stemmer by using the statistical unsupervised approach Instead of list corpus used.	Dictionary less approach.	Not dealing with Compound words.
		Based on the concept of n-grams.	Broken plural words are dealing properly.
[35]	Recurrent Neural Network (RNN) is based on the sequence to sequence encoding and decoding technique.	Statistical performance calculation was improved using the proposed model with the conditional probability of phrase pairs.	Needs multiple levels of linguistic regularities phrase level and word level.
[36]	Sequence to sequence framework proposed that converse to the next sentence prediction.	The model trained end to end and needless basic rules.	Lack of consistency.
[37]	Construction of lexicon for Urdu language using Bi-lingual dictionary.	Gain 86% of accuracy on Urdu website data set.	Need to add lexicon extension and sentiment scoring.
[28]	Text preprocessing and sentiment orientation by using movies and electronic appliances dataset.	Gain 82% accuracy in classification of text.	Needs to add more features.
			Needs to handle negation types properly.
[38]	Soundex and Shapex based Urdu spelling correction proposed.	Gain 94% accuracy on the corpus of 1.7 million words.	Similarity of the Shapex and Soundex spell correction needs to be improved.
[39]	Reverse edit distance technique was used to check spell checking.	Gain only 74% accuracy.	Transposition errors needs some more attention.
[40]	Word segmentation and lexicon-based Roman Urdu opinion mining is proposed.	Gain only 0.427 F-measure on the dataset of mobile phone user Roman Urdu reviews.	Needs to use noise detection techniques with word net or sentiment dictionary.
[41]	Proposing Urdu Name Entity Recognition system based on Bootstrap and CRF methods.	Gain 93% accuracy with space insertion and deletion techniques.	Need to explore the Urdu language further for the information retrieval process.
[42]	Supervised machine learning models Naive Bayes, K Nearest Neighbour and Decision tree is applied for the classification of Roman Urdu opinion sentiments.	Best results show Naive Bayes.	Needs to improve more results and need to apply more algorithms.

out how to address the issue of machine interpretation in the Urdu language is still in its beginning. Large datasets are a basic necessity for deep learning strategies to work and successfully modeled the assorted variety and catch the intrinsic unpredictability of language. The accessibility of the parallel corpora opens many ways for additional research for deep learning-based machine translation and transliteration of different linguistics. For example, [60] gives an equal corpus in 11 dialects including Dutch, English, Spanish, Swedish and Italian to give some examples. A parallel corpus for an enormous scope in the Urdu language is a road unexplored. Nonattendance of Roman-Urdu to Urdu Parallel is a bottleneck in investigating further exploration openings in the area of transliteration just as interpretation. CLE Pakistan [61] acquired a dataset of 100K Urdu words from education, health, business and training like different related areas. This is consists of two categories. One is imaginative and the other is informative with 20% and 80% ratio [62]. A large dataset created that contains 512,000 spoken words and 1,640,000 text words of Urdu. [63] has been done an incredible job by creating an Arabic Urdu script dataset contains Unicode characters and an XML format.

IV. MATERIAL AND METHODS

Transliteration is the method of transforming one language (Roman Urdu) into the target language (Urdu language) according to the exact pronunciation instead of focusing on meaning. It is a difficult task to develop a translator for the

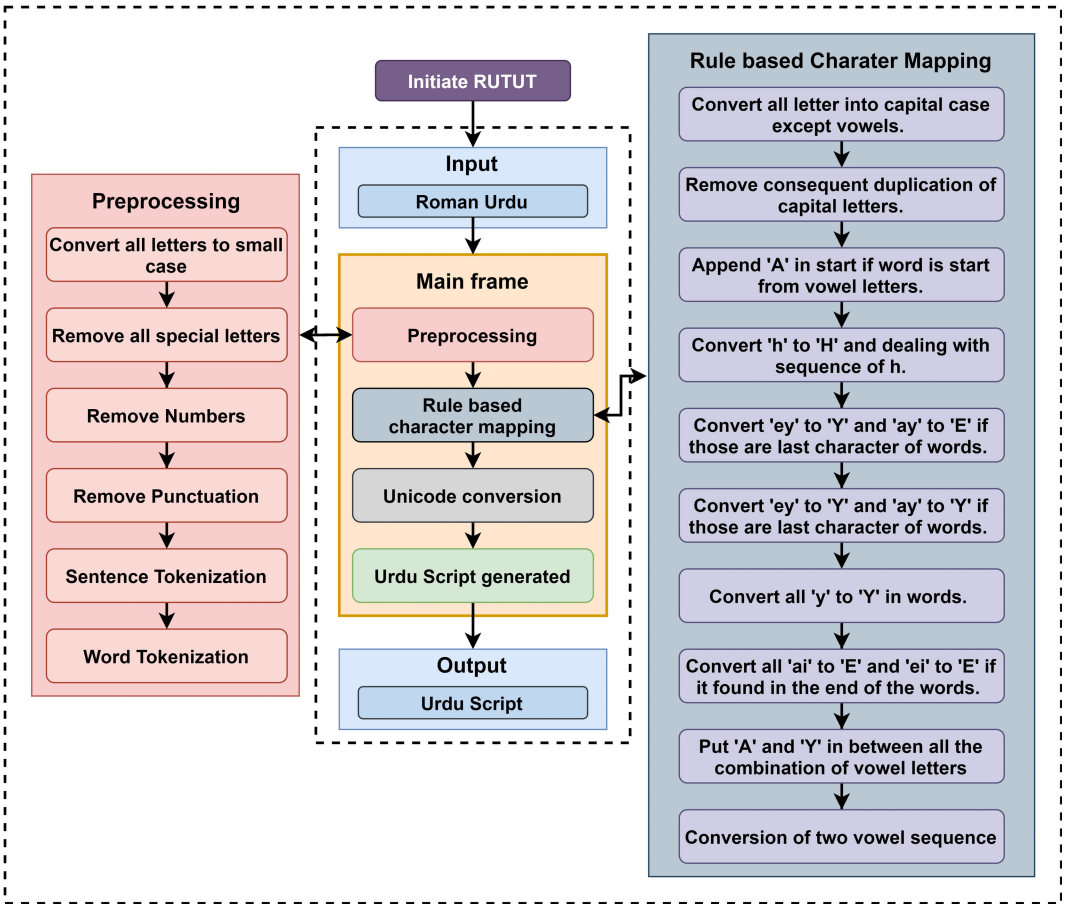


FIGURE 4. Architecture of the RUTUT: Rules-based character substitution and Unicode character mapping based Roman Urdu to Urdu Script Translator.

Roman Urdu language because it is not a standard language and does not have any proper rule of grammar or any writing vocabulary rules. A survey study [21] on mobile text data gives very interesting results that consist of the collection of 116 users and 346,455 mobile text messages. These results illustrate that a single word was typed by a single user with different spellings in different messages and also shows user try to complete the conversation in minimum words like using short forms. In this study, the main focus of the proposed approach is to develop a rule-based transliteration model for Roman Urdu that is the novelty which gives the proper standard to Roman Urdu. This section consists of the proposed methodology and methods that are used in this research to achieve its aims.

A. MAIN FRAME OF RUTUT

The proposed methodology is consists of three components such as preprocessing, rules-based character substitution and Unicode based character mapping shown in Figure 4. After initiating the RUTUT translator user can use user-friendly interface that depends on one input box, one output box and a translation button. When a user enters a Roman Urdu script as input then first pass this input to the first component preprocessing that filters the unnecessary data. After this, the preprocessed Roman Urdu script pass to the next component rule-based character substitution that consists of 12 different rules to convert any form of Roman Urdu into a specific form. The last component is the Unicode conversion that transforms the Roman Urdu characters into the Urdu as shown in Figure 17. After character by character Unicode mapping, the Roman Urdu completely transform into an Urdu script then this Urdu script can easily be used by the user. By using the proposed RUTUT translator the user can easily understand the exact meaning of the Roman Urdu script and can communicate to the other Roman Urdu user more expressively.

The RUTUT translator is developed for this research by using a Python programming language [64] in Anaconda platform in Jupiter notebook [65] and Tkinter library [66] is used to design Graphical User Interface (GUI) that makes it easy to use and user friendly.

B. PREPROCESSING

Roman Urdu text is in raw form, which needs to preprocessed. Reshaping of raw data by using data preprocessing methods [67], [68] is one of the data mining techniques. Translational models can learn from preprocessed data efficiently.

TABLE 2. Roman Urdu letters or sequences equivalent to Urdu vowels.

Roman vowels sequence	Urdu equivalent
a	Zabar
i	Zer
u	Paish
a, aa	Urdu alif
e, ei, ai, ay	Urdu bari-ye
i, ie, ee, ey	Urdu chooti-ye
oo, ou, au, u, o	Urdu vao

Real-world Roman Urdu data is incomplete, inconsistent, or missing in certain behaviors or trends, and is in all likelihood to incorporate many errors. Data preprocessing is an established approach to resolving such problems. In the world, the incompleteness of data is a general thing, lacking attribute values, errors, and outliers or containing only aggregate data. In preprocessing, firstly perform tokenization [69], [70] is the act of breaking up a sequence of strings into pieces such as words, keywords, phrases, symbols and other elements called tokens. Tokens can be individual words, phrases or even whole sentences. The word tokenization is applying to split the documents or sentences into individual terms that is helpful in filtering non-important words and punctuation. Second, in the case of sensitive system, capital case letter and lower case letter consider as different terms because of this conversion of capital case letters into lower case, reduce the unique terms in the documents. This increases the efficiency of the feature extraction process. Third, the process of changing statistics to something, a system can understand referred to as preprocessing.

C. RULE BASED CHARACTER SUBSTITUTION

Roman Urdu is not the standard language that has not any grammar rules and specific spelling or writing standards. That's why these rules are developed in this study which is the novelty. The Roman Urdu users use different spellings of a similar word in different messages. These rules transform the randomly spelt Roman Urdu script into one specific form that helps in further processing. Every step has its own importance. When preprocessed Roman Urdu as an input pass to the rule-based character substitution phase then it goes through the following steps:

1) RULE 1: CONVERSION OF THE VOWEL CASE

The preprocessed Roman Urdu needs to convert all the consonant letters into capital cases except 'y', 'h' and vowels ('a', 'e', 'i', 'o','u'). This mapping is applied to consonants only because vowels are very complex in one to one replacement processing. The preprocessing applied first that helps in mapping properly if any vowel letter accidentally remains capital then rules are not applied to this letter.

2) RULE 2: REMOVAL OF CONSEQUENT LETTERS

When all consonants become capital then some of the letters are consequently repeated in a capital case. Remove this duplication of letters from the capital case consonants

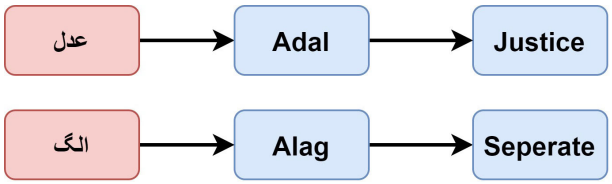


FIGURE 5. Impact of 'ain' and 'alif'.

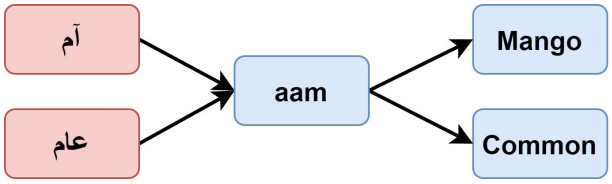


FIGURE 6. Impact of 'alif-mad' and 'ain'.

because of the germination of consonant letters written only once in Urdu. In the Urdu language, tashdeed is used as germination marks. Like 'ullo'(owl) in Urdu has a diacritical tashdeed sign because 'laam' is repeated but it was written once. In Roman Urdu 'ullo' has double 'l'. Same 'buddho'(stupid) has tashdeed sign on consonant 'daal'. It comes only once in the Urdu language with tashdeed but has double 'd' in 'buddho' in Roman Urdu. That's why to remove duplication of consequently capital letters because Urdu does not has any double letters. Roman Urdu has so many words that contain double letters but not in Urdu script.

3) RULE 3: INCLUSION OF LETTERS

The letter 'A' is included at the start of the word if any vowel letter found in the beginning. A vowel in Urdu script cannot occur at the start of the words without having before 'alif', 'ain', or 'hamza'. Mostly 'ain' and 'alif' occurs at the start of the word like 'adal' and 'alag' both have vowel letter 'a' in Roman Urdu that is equivalent to diacritic mark 'zabar' in Urdu as shown in Figure 5.

In the start of the vowels sequence 'aa' of the Roman Urdu word act as Urdu alif-mad that is equivalent to two consecutive Urdu 'alif' letters. The sequence 'aa' is equivalent to alif-mad or ain-alif like 'aam' target two Urdu script words as shown in Figure 6. A Roman Urdu word 'aur' has a sequence of vowels 'au' at the start. Table 2 shows that sequence 'au' equivalent to the Urdu word 'vao'. 'alif' written before 'vao' in Urdu script word 'aur' that shows the vowel rule at the start of the word is not applicable for 'a' vowel sound only.

4) RULE 4: REPLACE 'eh' AND 'oh'

The replacements that occur for the longest sequence from the left-hand side are as shown in Figure 7. An issue found in the transliteration of Roman Urdu to Urdu script was the vowel changed around Urdu 'gol-hay'. An example of the Roman Urdu word 'sheher'(city) shows that Urdu vowel 'zabar' present in between Urdu 'sheen' and Urdu 'gol-hay' and Urdu 'rey'. In pronunciation vowel 'e' places instead of 'a'.

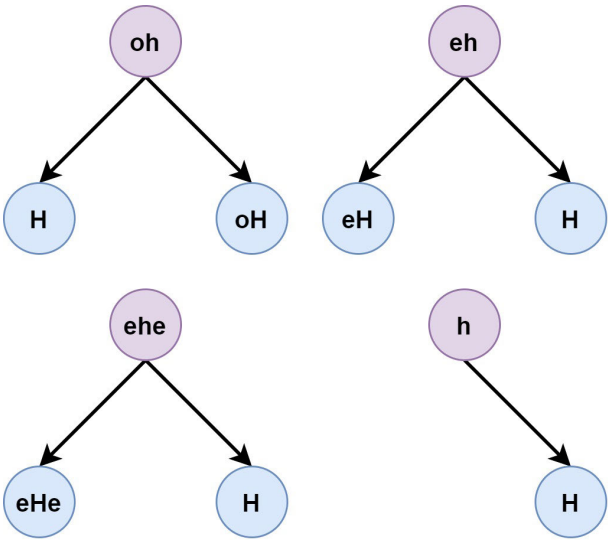


FIGURE 7. Substitution of 'oh' and 'eh' by capitalizing the 'h'.

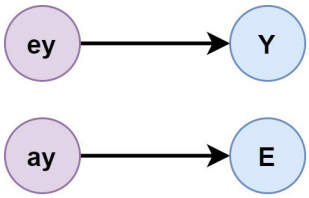


FIGURE 8. Substitution of 'eh' and 'ay'.

Similarly, a Roman Urdu word pronounces 'shohrat'(fame) has the Urdu vowel 'pesh' after 'sheen' instead of 'shuhrat'. This rule deals with vowels around 'gol-hay'.

- ehe = H, eHe
- oh = H, oH
- eh = H, eH
- h = H

5) RULE 5: REPLACE 'ey' AND 'ay' INTO 'Y' AND 'E'

Replace the 'ey' with 'Y' and 'ay' with 'E' if those are the last sequence of the Roman word that replacement is shown in Figure 8. If 'Y' found in the medial position of the Roman Urdu word then it is replaced by 'choti-ye' in Urdu script but it is replaced with Urdu 'bari-ye' if it is at the final position of the word. 'bari-ye' and 'choti-ye' produce different sounds when both are used as vowels. Roman Urdu word 'hain' contains an 'ai' vowel sequence in the middle of a word that sounds 'choti-ye' in Urdu. In the case of Roman Urdu word 'hai' contains the same sequence at the end of the word that sounds 'bari-ye' in the Urdu language as shown in Figure 9.

6) RULE 6: REPLACE 'ey' AND 'ay' TO 'Y' AND 'Y'

After rule 6 it also needs to replace the remaining 'ey' and 'ay' with 'Y' if 'y' is preceded with 'e' and 'a'. Figure 10 presents the replacements. 'y' acts as a consonant as well as a part of the vowel sequence.

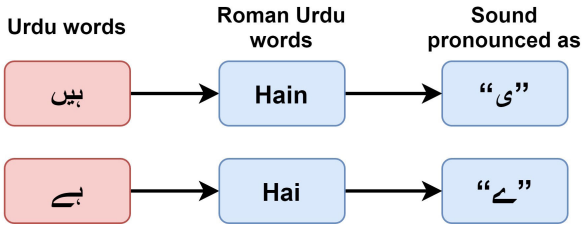


FIGURE 9. Substitution of 'ai' effects in Urdu words.

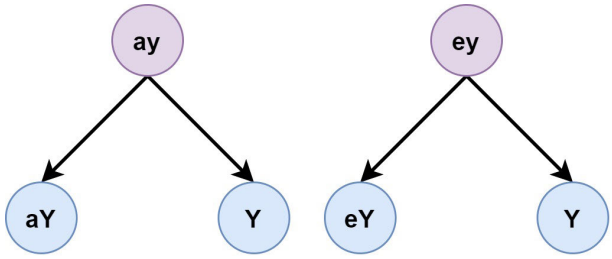


FIGURE 10. Substitution of 'eh' and 'ay' by capitalizing 'y'.

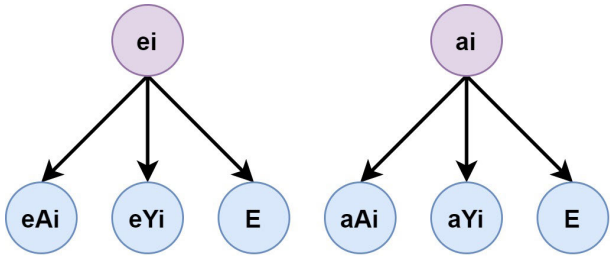


FIGURE 11. Substitution of 'ei' and 'ai'.

7) RULE 7: CONVERSION OF 'y'

Convert the remaining 'y' into the capital case. All special cases of 'y' already handled in previously defined rules. Now, this replacement is generally placed to handle the remaining 'y'.

8) RULE 8: CONVERSION OF 'ai' AND 'ei'

if the vowel sequence 'ai' and 'ei' present at the end of the word then done the following replacements shown in Figure 11. Roman vowels character are considered as a special case of syllable boundary when it appears in a sequence. Syllable boundaries cannot be predicted in Urdu and not in Roman Urdu script. If consonants appear before vowels in the syllable then general rules are enough to translate it but if vowels appear in the start then Rule IV-C3 applies to it. Table 2 shows the Roman script that a single Urdu vowel character can be a map on a single Roman Urdu character or two-character sequence. A sequence of Roman Urdu characters can be considered as equivalent to one or two Urdu vowels. The 'ai' is the two vowel sequence corresponding to a single letter in Urdu 'bari-ye' or belongs to two different syllables. The start of the second syllable is 'i' and 'a' belongs to the first syllable. 'A' and 'Y' are introduced in the representation of Urdu 'hamza' and 'ain' because these are needed before the vowels.

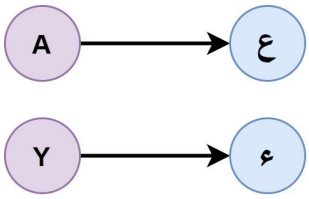


FIGURE 12. Substitute 'A' and 'Y' in between the vowels.

9) RULE 9: HANDLING VOWEL COMBINATION

Find all the combinations of vowels character sequence if found any sequence of two or more vowels and place A for (ain) and Y for (hamza) in between them. This is the generalized form of Rule IV-C8 that deals with all possible interpretations of the vowel sequence. As the example shown in Table 2 and Figure 11 'a-i' and 'ai' has two vowel combinations of the 'ai' vowel sequence. Then for further processing 'aAi', 'aYi' and 'ai' are generated. In another vowel sequence 'ua' has only a 'u-a' combination. This does not map on any single Urdu vowel. That's why another combination is not valid and 'uYa' and 'uAa' are generated for the next processing. In vowel sequence 'aai'(she came) has three combinations 'aa-i', 'a-ai' and 'a-a-i'. 'aAai', 'aaAi', 'aYai', 'aAaAi', 'aaYi', 'aYaYi', 'aAaYi', and 'aYaAi' combinations are generated for further processing after applying this rule.

10) RULE 10: CONVERSION OF TWO VOWEL COMBINATION

It is generally a one to one character mapping rule that defines the way of encoding against vowel sequences in the Roman Urdu script. The encoding scheme of the Roman Urdu vowel sequences shown in Figure 13.

11) RULE 11: SUBSTITUTE THE VOWELS AT THE FINAL POSITION

Search the given below vowels at the end of the Roman Urdu words and perform the following substitution.

- e = E
- i = Y
- a = A, H
- u = O

Word's last character has always a long vowel in Urdu scripts like in Roman Urdu word 'aadmi'(man) 'i' present at the final position that is not ambiguous between the long vowel (chotiye) and short vowel (zer). Roman Urdu 'sada'(simple) 'a' at the final position but in Urdu script 'a' is substituted with 'gol-hay'.

12) RULE 12: GENERALIZED CONVERSION OF VOWELS

Replace the remaining vowels as given below. The generalized form of the rule comes forward after all the special rules. This is the last rule of the rule-based character mapping.

- a = A, null
- e = E
- i = Y, null
- o = O
- u = O, null

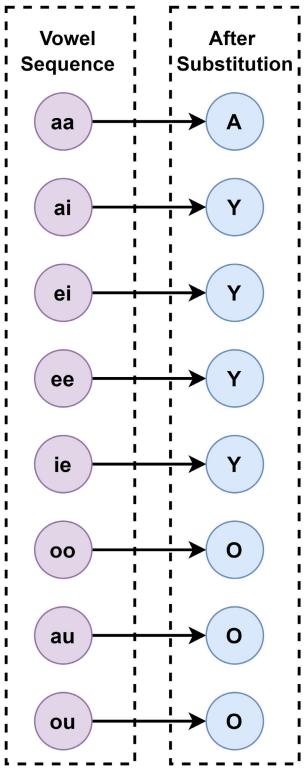


FIGURE 13. Presenting replacements of vowels sequence with capital letters.

D. URDU UNICODES

The Urdu language is inscribed in calligraphic Nastaliq script that has 39 to 40 unique letters. Simply, the Urdu translates into the Latin alphabets then it is known as Roman Urdu that eliminates many linguistic pronunciation elements that are not found in any equivalent in the English language but in Latin script or other languages are used them in writing. The Persio-Arabic script in the modified form is named Urdu script. These are close to the phase of the Nastaliq style of Perso-Arabic script development. In 1911, the Urdu typewriter invented then Urdu newspapers start publishing the handwritten scripts by katibs or khush navees (calligraphers) until the 1980s. The Daily Jang was the first Pakistani national newspaper that composed and published the newspaper on the bases of computer-based Nastaliq. There are so many efforts underway that focus on the development of the more user-friendly and sophisticated Urdu linguistic support system on the internet and computers. In this modern age, nearly all Urdu the journals, magazines, periodicals and Urdu newspapers are all composed in the computer by using Urdu phonetic based software.

1) UNICODE

The Unicode based communication is the general standard of character encoding pattern utilized for the describing text for machine processing. It gives further knowledge about the characters and their application. It presents uniform methods of encoding and decoding the multi-linguistic textual data and brings order to the phase of operations that made it

ا A u0627	آ AA u0622	ب B u0628	پ P u067E
ت T u062A	ج J u062C	س S u0633	چ CH u0686
ہ H u062D	خ KH u062E	د D u062F	ذ Z u0632
ر R u0691	ش SH u0634	غ GH u063A	ف F u0641
ک K u06A9	گ G u06AF	ل L u0644	م M u0645
ن N u0646	و O u0648	ی Y u0649	ے E u06D2

FIGURE 14. Unicode chart on the bases of character mapping rules.

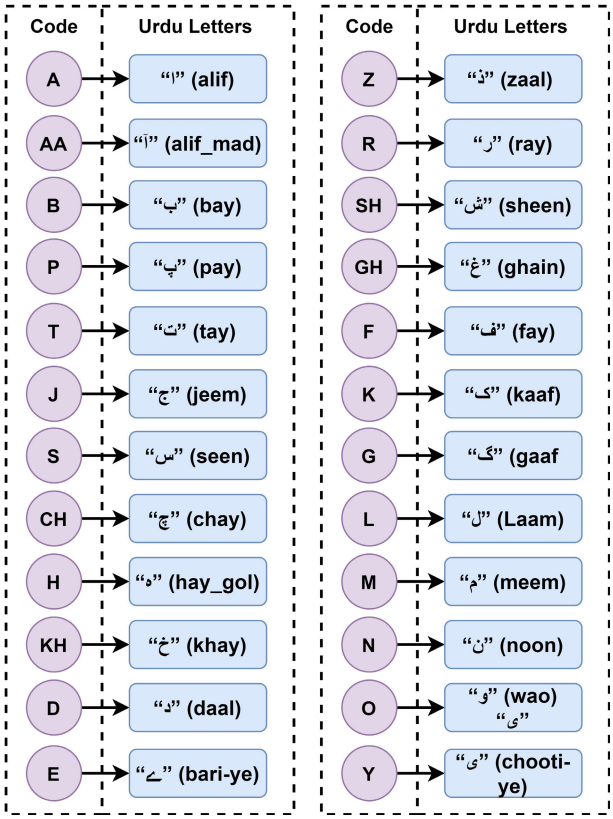


FIGURE 15. Mapping sequence of Roman Urdu to Urdu character.

hard to trade text data globally. Researchers who deal with multi-linguistic scripts in the Urdu language in the computer system like in business, researchers, scientists and linguists others also discover that it makes their work simple. The scheme of Unicode is based on the flexibility and simplicity of ASCII but goes considerably ahead from ASCII's insufficient capability to encode only significant Latin script.

All written languages of the world can be efficiently encoded in Unicode standard's capabilities. It assigns the numerical value and name to each character that makes it simple and efficient for linguistic processing. Three encoding forms supported by the Unicode standards are UTF-8, UTF-16 and UTF-32 that have a common repository of characters. Those encoding methods support for encoding millions of characters. This is enough for all associated character encoding conditions, including full coverage of all historic scripts of the world and also for common notational affairs. The Unicode standard specifies codes for characters practiced in all the influential kinds of literature written today. The scripts cover the Middle Eastern right-to-left scripts, European alphabetic scripts, and many other scripts of Asia. More than 135,000 characters codes present by the Unicode standard from the world's alphabets, collections of symbols and writing systems. Unicode considers the purpose of giving a code point (a number) that is unique for each character in text processing, not a symbol for each character. In another way, Unicode describes characters in an abstract form and gives the visual rendering (style, shape, size or font) to the machine program or software like the web browser or word processor. The simple aim of the Unicode designers becomes complex when concessions are made in the hope of promoting the Unicode system rapidly. The Unicode chart is shown in Figure 14 which represents the Unicode for the characters of the Urdu script. These Unicode are used in this study to encode or map the Roman Urdu characters to Urdu characters.

2) CHARACTER MAPPING

Today computer generation needs to understand the Roman Urdu or any other language. when it was about the machine or computer then it became necessary that computers can understand every single word or character properly. The computer does not recognize the shape or character of the language. Every language has its style and shape of the alphabet that makes it difficult to understand. The computer machine working on the bases of the codes (are numerical values) that can easily understandable for the computer. With the rapid growth of the internet and digital media, it became necessary to develop a system of codes for languages or codes for each style or shape of the character. Then the Unicodes system developed to recognize the language alphabets for the computer [71], [72].

In the proposed RUTUT approach, the scheme of character mapping is shown in Figure 15. Previous section IV-C discussed issues in the translation of Roman Urdu to Urdu and its solution. By analyzing those issues develop this scheme of Unicodes for the Urdu alphabets. Character mapping is the phase in which a preprocessed and rule-based substitutions are applied to Roman Urdu and a form of Roman Urdu is

obtained. After those phases, this Roman Urdu form needs to translate into the Urdu Language. For this purpose, the Unicode list used to substitute Roman Urdu with Urdu script. The Unicode based character map shown in Figure 14 and scheme of substitution is shown in Figure 15. When all of the character mapping rules and substitution into Urdu script using Unicode are applied on the input (Roman Urdu) script then the Urdu form comes as an output. The computer system translates the Roman Urdu character alphabet by alphabet into Urdu Unicodes that presents as Urdu script as an output.

V. RESULTS AND DISCUSSION

In this research, Roman Urdu to Urdu script translational (RUTUT) model is developed as shown in Figure 19 consists of rule-based character substitution and Unicode based character mapping techniques. Those are fundamental techniques to translates one language into another. Every language has its own grammar, spelling standards and rules to write but Roman Urdu is not the standard language. Roman Urdu language has not a specific rule to spell words. When the user write one word in different spellings with different capitalizations in words then it is a difficult task to recognize the pattern of the words. This is a very necessary task to recognize the fundamental patterns in any language that gives it a proper meaning. By analyzing different issues in translating the Roman Urdu, this research solves those issues one by one as discussed in section IV-C. As solution to those issues rules are building up one by one which gives the pattern of the words of Roman Urdu structures.

Algorithm 1 RUTUT

```
System Initialization
procedure Mainframe(input)
    for sentences in input do
        filter = Call Preprocessing(sentences)
        for Words in filter do
            Substitution = Call CSR(Words)
            SubSent = appending romanized substituted words.
        for Words in SubSent do
            TranslatedWords = Call UCM(Words)
            TranslatedSent = append TranslatedWords
        Output = Appending translated sentences
    Return Output
```

A. RESULTS

The focus of this research is to propose the fundamental rules that will help to transliterate Roman Urdu into Urdu script as well as gives the standard rules to evolve the Roman Urdu. In this section, RUTUT translator evaluation is presented in different ways and comparing it with two different translators. The proposed model RUTUT translator's structure is consists of three phases which present how the RUTUT structure converts the Roman Urdu into Urdu script. Gives outstanding results that are discussed in upcoming sections.

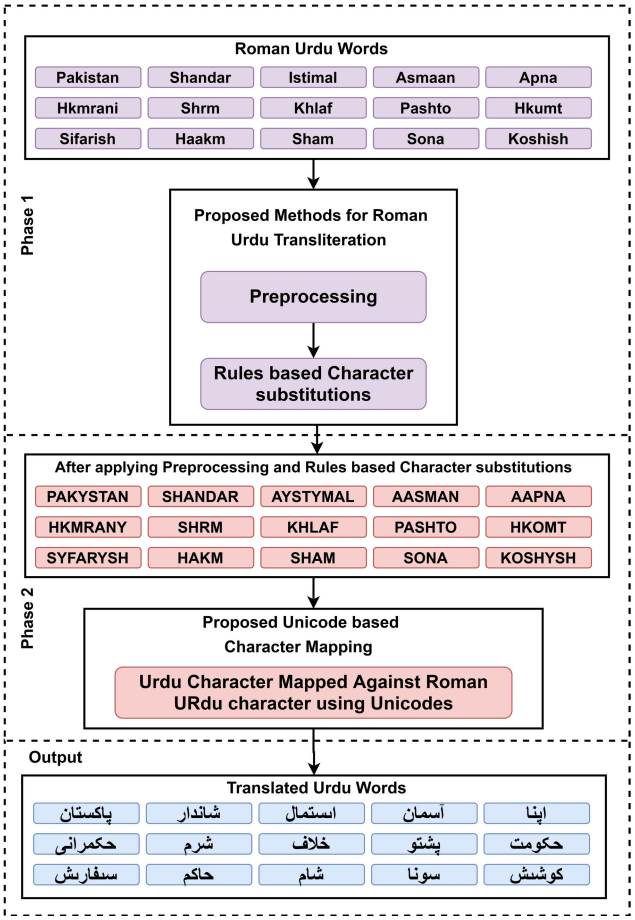


FIGURE 16. Step by Step Roman Urdu Word Translate into Urdu Words.

B. RESULTS OF PROPOSED MODEL RUTUT TRANSLATOR

RUTUT translator consists of three phases that present the step by step results in Figure 16. Phase 1 consists of two modules such as preprocessing and rule-based character substitution technique. A Roman Urdu script enters as an input to the first phase in its raw form that contains impurities which can reduce the accuracy of the transliteration results. There are also specific steps such as capital to lower case conversion, remove the special symbol, removal of numbers, sentence tokenization and word tokenization are performed as shown in Figure 4. In the second module, rule-based character substitution applied on that filtered form of the preprocessing phase as shown in Figure 16. This module consists of 12 basic rules which substitute character against character. After applying these steps of two modules a filtered Romanized form comes as an output.

The output of the phase first becomes an input of the second phase. Then the Unicode based character mapping scheme is designed in this phase. This scheme of mapping is based on Unicode that is the four digit code. By using this code of scheme each character of Roman Urdu mapped

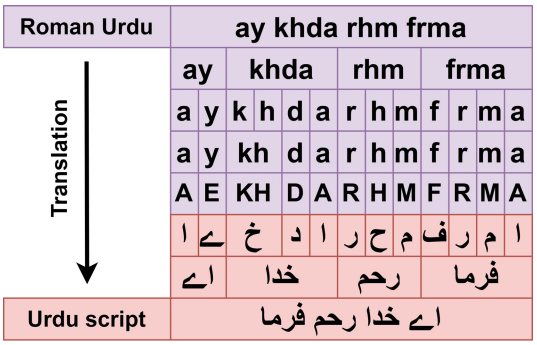


FIGURE 17. Character by character translation.

against Urdu character. When replaced each character with Unicodes then it automatically converts into an Urdu language. Computer machines are all working on the functionality of Unicode based linguistic. Then those converted Urdu characters create words and sentences. Figure 16 presents the results of each phase and sequence of the work flow of the RUTUT translator. There are 15 Roman Urdu words gives to the RUTUT translator as input then it converts those words into a filtered Romanized form. This Romanized form of words shown in the first block of the second phase. There are 15 filtered romanized words that are all in a capital case. These words were developed by the above preprocessing and rules-based character substitution modules. The second phase perform the Unicode based character mapping on this output of the First phase then successfully obtain Urdu translation of the 15 Roman Urdu words. All of the 15 words translated very perfectly by using the RUTUT translator.

C. RESULTS OF CHARACTER BY CHARACTER TRANSLATION

The character by character translation presents in Figure 17. There is translation performed on a Roman Urdu sentence ''ay khda rhm frma'' and transform this sentence into its Urdu form.

First of all, preprocessing is applied to this sentence then split each word by using word tokenization such that '' 'ay','khda','rhm','frma' ''. After this, rules based character substitution is applied on these words that further divide the words into letters and transform in Romanized Filtered form such that '' 'A','E','KH','D','A','R','H','M','F','R','M','A' ''. Then Unicode based character mapping applies to this form that transforms those letters into Urdu form and joins them to create Urdu words or sentences. Roman Urdu language is the one form of Urdu that is written by using English letters. That's why Roman Urdu is right-hand side language that starts from the right-hand side and ends at the left-hand side.

When all letters are accurately mapped on the Urdu letters and Urdu form obtained then because of Unicode character mapping it automatically starts from the left-hand side. Urdu is the left-hand side language. After this a full sentence of proper formatted Urdu script is obtained. At the end, Urdu

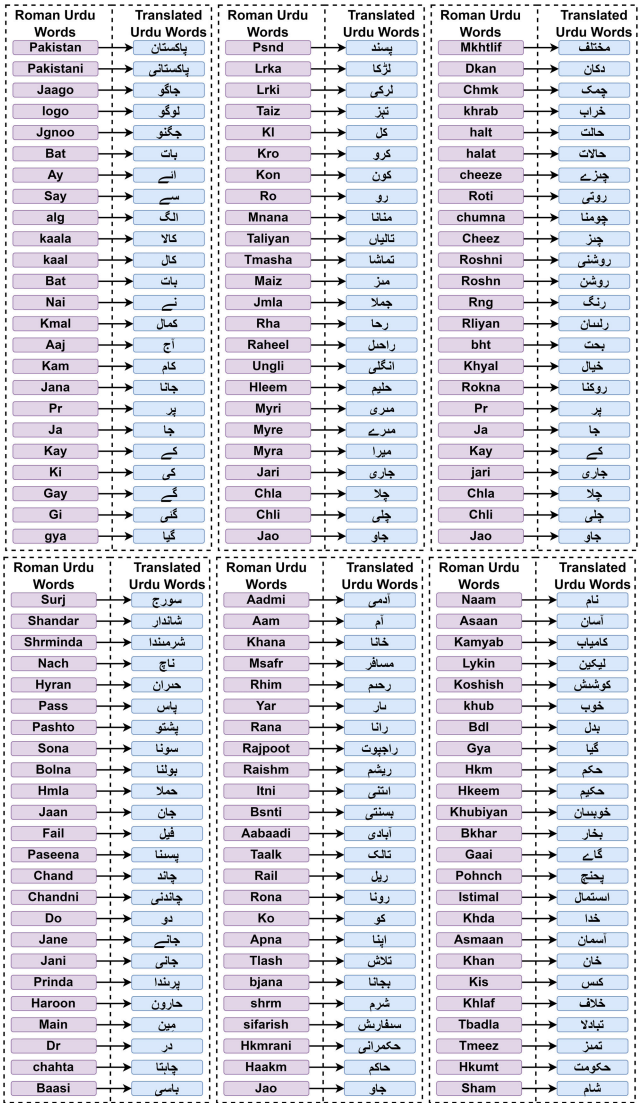


FIGURE 18. Word based Roman Urdu script translation using RUTUT.

script is obtained in same sequence as the Roman Urdu Words. The character by character translation process is presented in Figure 17 that shows how the translation process is done efficiently.

The performance of the proposed RUTUT translator needs to evaluate. For the evaluation process, two methods are used in the paper. First, adopt the fundamental approach in which 2000 Roman Urdu words are used as an input to the RUTUT translator. From which 1917 Roman Urdu word accurately translated into the Urdu language that shows the RUTUT translator 95.8% accurately translates the Roman Urdu words into the Urdu Language as shown in Table 3. The 144 Roman Urdu words as an example shown in Figure 18, are evaluated using RUTUT translator. From 144 Roman Urdu words, about five words are miss spelt in Urdu form and 139 words are accurately translated that shows 96% accurate transliteration results. Second, the comparison approach adopted to evaluate further that discussed in the next section.

FIGURE 19. RUTUT's Tkinter GUI that presents results of translations performed by RUTUT.

Roman Urdu Input	Google Translator Result 1	Google Translator Result 2	ijunoon Roman Urdu to Urdu Translator	RUTUT Proposed Translator
surj chand khda kay bnae hue hain	سورج چاند کھڈا کیا بنے ہو ✗	سورج چاند سے بنا ہے ✗	سورج چند خدا کے بنائی ہوئے ہیں ✗	سورج چاند خدا کے بنائے ہوئے حسن ✗
ay pakistan kay logo jago	آے پاکستان کا لوگو جاگو ✗	آے پاکستان کا لوگو جاگو ✗	اے پاکستان کے لوگوں جاگو ✓	اے پاکستان کے لوگو جاگو ✓
pakistani roshn khyal log hain	پاکستانی روشین خیال لاگ ہے ✗	پاکستانی روشن خیال افراد ✗	پاکستانی روشن خیال لوگ ہیں ✓	پاکستانی روشن خیال لوگ حسن ✗
shkr hai khda ka	shkr ہے کھڈا کا ✗	شوگر کھڑی ہے ✗	شکر ہے خدا کا ✓	شکر ہے خدا کا ✓
mehnt krna lazmi hay	mehnt krna lazmi hay ✗	مشکل کام ✗	میحنت کرنا لازمی ہے ✗	محنت کرنا لازمی ہے ✓
ap ki baat drst nhi hay	نہیں ہے drst آپ کی بات ✗	تم خوفزدہ نہیں ہو ✗	آپ کی بات درست نہیں ہے ✓	آپ کی بات درست نہیں ہے ✓
kis ki tlash main ho	کس کی تولاش مین ہو ✗	تم کیا تلاش کر رہے ہو ✗	کس کی تلاش میں ہو ✓	کس کی تلاش میں ہو ✓
insaan kay bche bno	ایران کیا بیچے بنو ✗	انسان ہو یا بچ جانے والا ✗	انسان کے بچے بنو ✓	انسان کے بچے بنو ✓
to sb log tlash main lg gay	to sb log tlash main lg gay ✗	تو سب کی تلاش شروع ہوگئی ✗	تو سب لوگ تلاش میں ایل جی گے ✗	تو سب لوگ تلاش میں لگ گے ✓
dost sno	dost sno ✗	کافی نیند ✗	دوست سنو ✓	دوست سنو ✓
ay khda rhm frma	ay Khuda rhm فرما ✗	آنے خدا rhm fram ✗	اے خدا رحم فرما ✓	اے خدا رحم فرما ✓
Kya Main Aapki Mdad Kr Sakta Hun	کیا میں آپکی معداد کر سکتہ ہن ✗	کیا میں اپ کی مدد کر سکتا ہوں ✓	کیا میں آپکی مداد کر سکتا ہوں ✗	کیا میں آپکی مدد کر سکتا ہوں ✓

FIGURE 20. Comparison of the proposed RUTUT translator with ijunoon and Google online translator where 'X' presents the sentence contain mistakes in transliteration and '✓' presents accurately translated sentences.

The RUTUT translator is shown in Figure 19 that consist of user friendly GUI. First, user needs to enter Roman Urdu script as an input and then press the translate button. The Roman Urdu script goes through three major modules

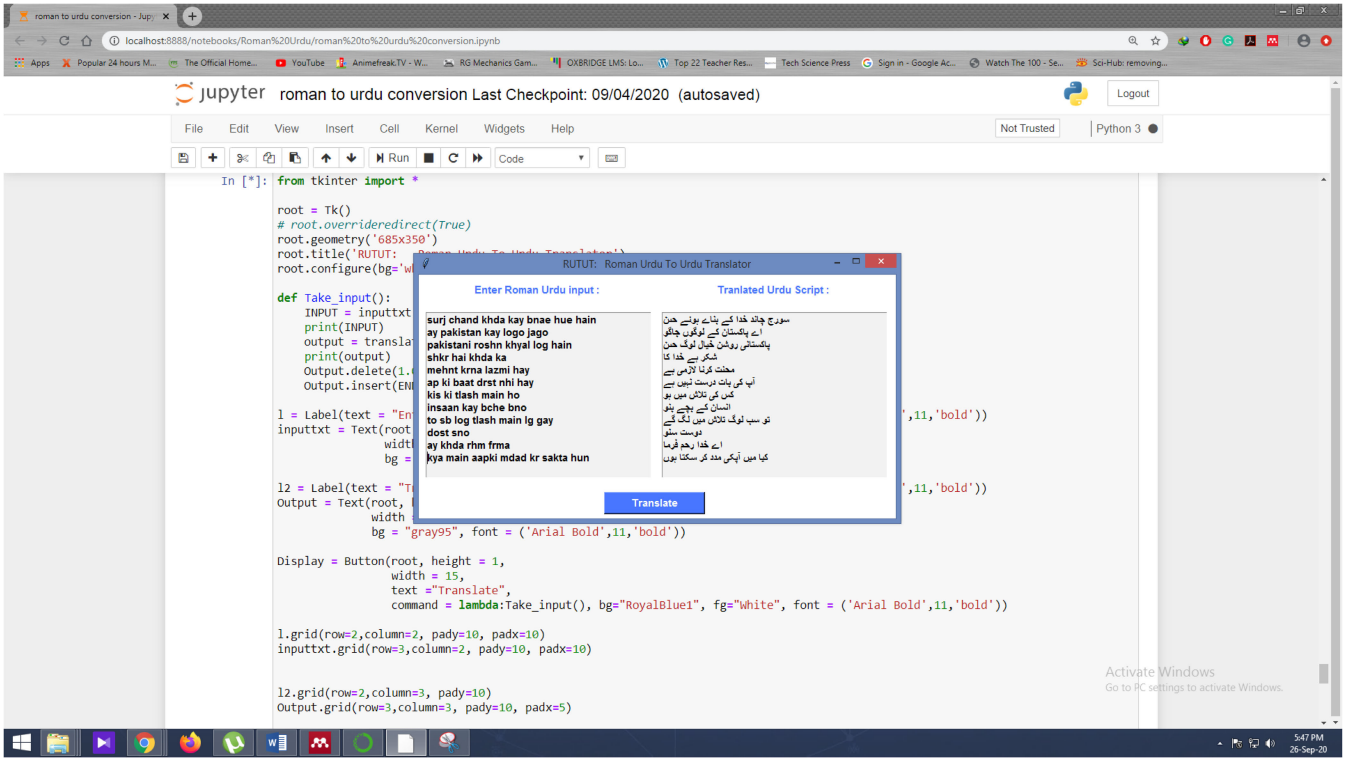


FIGURE 21. Transliteration results of the proposed approach RUTUT.

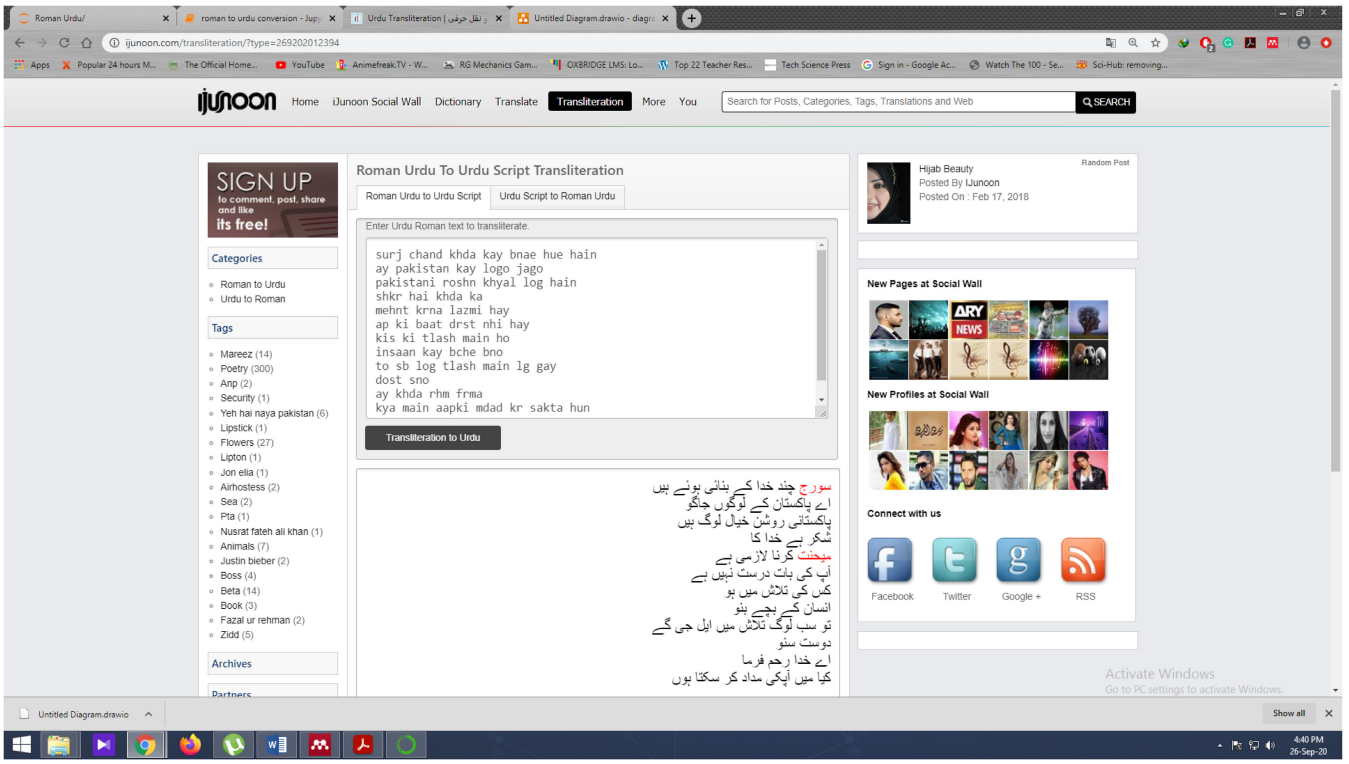


FIGURE 22. Transliteration results of the ijunoon online translator.

preprocessing, rule-based character substitution and Unicode based character mapping as shown in Figure 4 and 16. Figure 19 presents the translation of the 12 sentences that are further compared with ijunoon and google online translators.

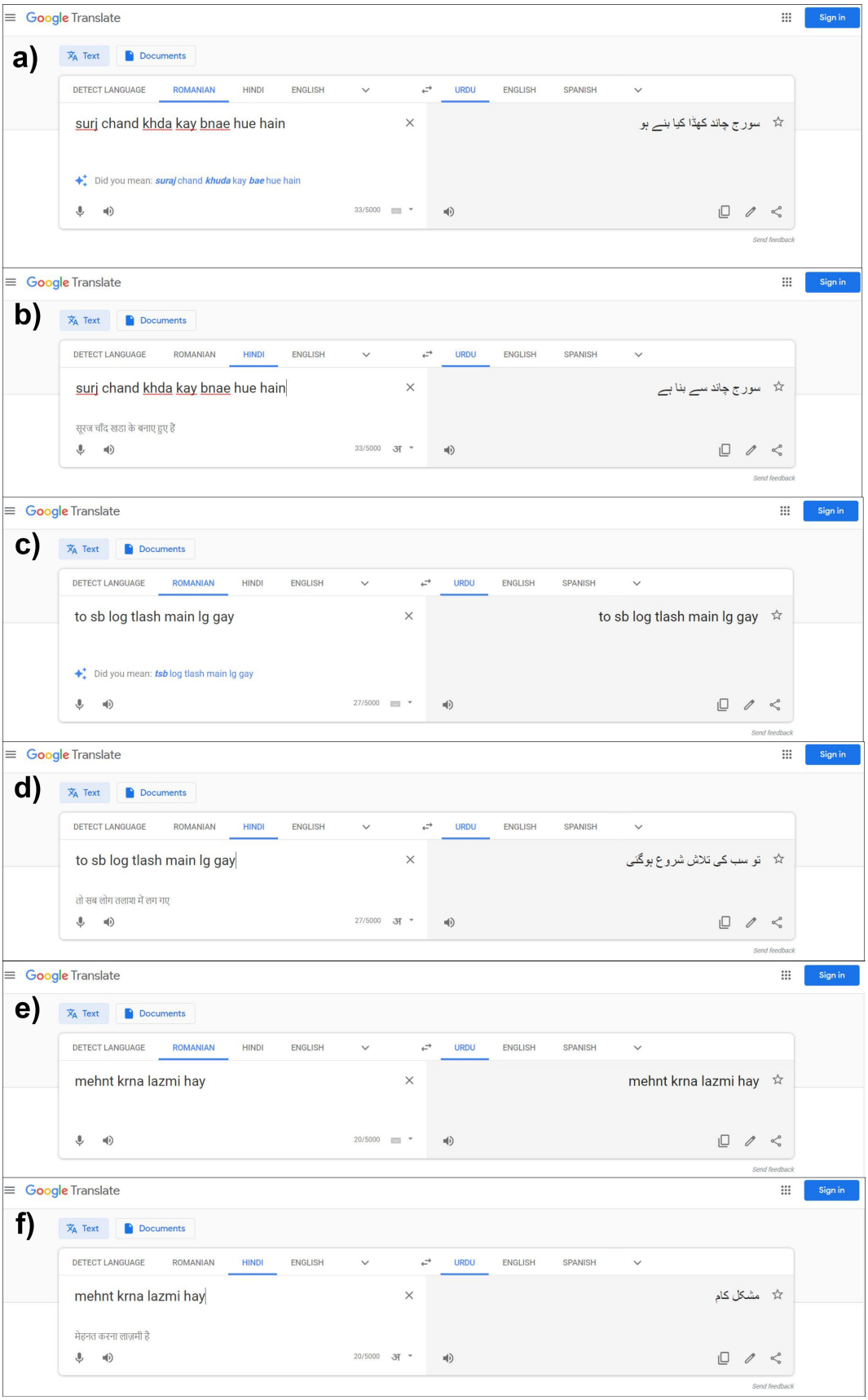


FIGURE 23. Google online translator results after detection of Roman Urdu as a Romanian and Hindi language.

TABLE 3. Transliteration Results based on Roman Urdu words.

Roman Urdu words	Results
Total Words	2000
Correctly Translated	1917
Incorrect Translated	83
Correctly Translated Percentage	95.8%

VI. COMPARISON OF THE EXISTING AND PROPOSED RUTUT TRANSLATOR

In this research, the proposed approach RUTUT compares to the ijunoon [73] and google [74] online translator. Roman Urdu to Urdu script translators is already very few in this world. Many authors present their work on Roman Urdu to English translation but are very few for Urdu translation. Figure 20 consists of five columns which contain Roman Urdu input sentences, Google translator result 1, Google translator result 2, ijunoon translator results and RUTUT translator results. The twelve sentences are randomly selected to test the translator results and for their comparison with the proposed approach.

A. COMPARISON OF THE GOOGLE ONLINE TRANSLATOR RESULTS

First, the Google translator [74] results are illustrated in this figure because Google is from one of the largest industry in the world that has a Google online translator for language translations. But the Google translator does not have any specific translator for the Roman Urdu because it is not the standard language. The Google translator relies on grammatical or translational rules of the other languages. When the user enters the Roman Urdu input then it automatically detects it as a Romanian or Hindi language and translates it into Urdu according to the rules of these languages. That's why there are two Google translator result1 and result2. Like in the first sentence 'surj chand khda kay bnae hue hain' is translated into 'Google Translator Result1' that illustrates the google translator auto-detect the Roman Urdu script as a Romanian language and translate it into Roman Urdu according to the Romanian language rules. Second 'Google Translator Result2' auto-detect the Roman Urdu script as a Hindi language and translate it into Roman Urdu according to the Hindi language rules. Figure 20 presents that Google translator gives poor results in both outputs. It translates the Roman Urdu script into the Urdu sentence but not according to the transliterational concepts. It changes the meaning as well as phonetics of the language and does not perform translation for some sentences as shown in the column 'Google translator result1' sentence number 4, 5, 9, 10 and 11. Further, 'X' presents false or misspelt translations in Figure 20 and '✓' present correct translations without any single mistake. Only one sentence in the last of the second column 'Google translator result2' that depends on the rules of Hindi language, is correctly translated. Figure 23 displays the results of some sentences in which part a and b presents the same sentence input but translated into different sentences according to Romanian and Hindi language rules. Same like this part c and d, e and f are also present different results.

TABLE 4. Comparison of the existing and purposed approach.

Sr.#	Findings	Neural network based translit-eration system [35], [75]	Sequence to Se-quence de-coding and en-coding based translit-eration system [32], [36]	Translite-ration based on heuristic parse tree [37], [76]	RUTUT Pur-posed Ap-proach
1	Rule based charac-ter substitution.	NO	NO	NO	YES
2	Preprocessing based filtration of text.	YES	NO	YES	YES
3	Unicode based character mapping.	NO	NO	NO	YES
4	Perform fast trans-lations.	YES	NO	YES	YES
5	User friendly GUI.	NO	NO	NO	YES
6	Easy to implement in real life sce-nario.	NO	YES	NO	YES
7	Provide standards to Roman Urdu language.	NO	NO	NO	YES
8	Accurately trans-lational results.	80%	48%	86%	95.8%

B. COMPARISON OF THE IJUNOON ROMAN URDU TO URDU TRANSLITERATION RESULTS

The ijunoon [73] online Roman Urdu to Urdu translator present the transliterational results according to exact pronunciations of the words. The 3rd column of the Figure 20 shows the results of the ijunoon translator with the same twelve sentences as used in the comparison. The ijunoon and proposed RUTUT translator gives better results than the Google online translator. The ijunoon translate all twelve sentences into the Urdu sentences but having mistakes in sentence number 1, 5, 9 and 12. But in comparison, the Proposed approach RUTUT translator only having a mistake in sentence number 1 and 3. The RUTUT translator gives better results than the ijunoon translator.

C. COMPARISON RESULTS OF PROPOSED APPROACH RUTUT: ROMAN URDU TO URDU TRANSLATOR

The last column of Figure 20 presents the results of the proposed approach RUTUT translator. Figure 20 and discussion of previous comparison sections, clearly shows that the google translator is not able to translate the Roman Urdu script into the Urdu language because the google translator does not have any standard rules and grammar for the Roman Urdu script. The RUTUT translator translates all the sentences into accurate sentences according to the meaning and

words. Twelve sentences translated by the RUTUT translator and 10 sentences are correctly translated without any mistakes. Only two sentences, number 1 and 3 got minor mistakes. The ijunoon is also shown mistakes in four sentences but RUTUT shows mistakes in only two sentences.

These results clearly show that the RUTUT translator gives much better results than Google and ijunoon online translators because of its basic rules that are developed in this research to translate the Roman Urdu into Urdu script. These are the most latest online translator on the internet that does not have any specific rules and standards for Roman Urdu translator because Roman Urdu is not the standard language.

Further, to evaluate the proposed research, Table 4 shows the comparison of the existing system with the proposed RUTUT translator that illustrates the novelty in terms of findings and evaluates that the RUTUT translator outperforms the existing systems in comparison.

VII. CONCLUSION

The RUTUT, a proposed translator translate the Roman Urdu script into Urdu script by using preprocessing, rule based character substitution and Unicode based character mapping techniques.

- At the initial stage, when the user gives a Roman Urdu script as an input then preprocessing rules are applied that filter unnecessary data. The rule-based character substitution process converts the filtered form of Roman Urdu into a single standard form. The character mapping module based on Unicodes of Urdu characters against the Roman Urdu characters that convert the Roman Urdu form into Urdu Script.
- The evaluation of the proposed model presented by translating 2000 Roman Urdu words into Urdu in which 1917 words are accurately spelt that shows the 95.8% words are accurately translated. Furthermore, the RUTUT results are compared with the Google online translator and evaluated based on several sentences.
- This research successfully achieves its aims by developing the translational rules for the Roman Urdu script and by developing a RUTUT translator based on these rules that translates Roman Urdu into Urdu script.

REFERENCES

[1] P. Arora, D. Shterionov, Y. Moriya, A. Kaushik, D. Dzendzik, and G. Jones, "An investigative study of multi-modal cross-lingual retrieval," in *Proc. Workshop Cross-Lang. Search Summarization Text Speech (CLSSTS)*, 2020, pp. 58–67.

[2] J. Capstick, A. K. Diagne, G. Erbach, H. Uszkoreit, A. Leisenberg, and M. Leisenberg, "A system for supporting cross-lingual information retrieval," *Inf. Process. Manage.*, vol. 36, no. 2, pp. 275–289, Mar. 2000.

[3] S. Mukund, R. Srihari, and E. Peterson, "An information-extraction system for Urdu—A resource-poor language," *ACM Trans. Asian Lang. Inf. Process.*, vol. 9, no. 4, pp. 1–43, Dec. 2010.

[4] K. Riaz, "Baseline for urdu IR evaluation," in *Proc. 2nd ACM Workshop Improving Non English Web Searching (iNEWS)*, New York, NY, USA, 2008, p. 97.

[5] M. Humayoun, H. Hammarström, and A. Ranta, "Urdu morphology, orthography and lexicon extraction," in *Proc. 2nd Workshop Comput. Approaches Arabic Script-Based Lang.*, 2007, pp. 6–40.

[6] K. Riaz, "Rule-based named entity recognition in Urdu," in *Proc. Named Entities Workshop*, 2010, pp. 126–135.

[7] F. Adeeba and S. Hussain, "Experiences in building urdu wordnet," in *Proc. 9th Workshop Asian Lang. Resour.*, 2011, pp. 31–35.

[8] M. Shahroz, M. F. Mushtaq, M. Ahmad, S. Ullah, A. Mehmood, and G. S. Choi, "IoT-based smart shopping cart using radio frequency identification," *IEEE Access*, vol. 8, pp. 68426–68438, 2020.

[9] M. J. H. Mughal, "Data mining: Web data mining techniques, tools and algorithms: An overview," *Int. J. Adv. Comput. Sci. Appl.*, vol. 9, no. 6, pp. 208–215, 2018.

[10] S. Bhatia, P. Chaudhary, and N. Dey, *Opinion Mining in Information Retrieval*. Singapore: Springer, 2020.

[11] K. Mehmood, H. Afzal, A. Majeed, and H. Latif, "Contributions to the study of bi-lingual roman urdu SMS spam filtering," in *Proc. Nat. Softw. Eng. Conf. (NSEC)*, Dec. 2015, pp. 42–47.

[12] N. Khan, M. P. Bakht, M. J. Khan, A. Samad, and G. Sahar, "Spotting urdu stop words by Zipf's statistical approach," in *Proc. 13th Int. Conf. Math., Actuarial Sci., Comput. Sci. Statist. (MACS)*, Dec. 2019, pp. 1–5.

[13] W. Khan, A. Daud, K. Khan, J. A. Nasir, M. Basheri, N. Aljohani, and F. S. Alotaibi, "Part of speech tagging in urdu: Comparison of machine and deep learning approaches," *IEEE Access*, vol. 7, pp. 38918–38936, 2019.

[14] T. Ehsan and S. Hussain, "Analysis of experiments on statistical and neural parsing for a morphologically rich and free word order language urdu," *IEEE Access*, vol. 7, pp. 161776–161793, 2019.

[15] E. T. Al-Shammari and J. Lin, "Towards an error-free Arabic stemming," in *Proc. 2nd ACM Workshop Improving Non English Web Searching*, 2008, pp. 9–16.

[16] B. Jawaid and T. Ahmed, "Hindi to Urdu conversion: Beyond simple transliteration," in *Proc. Conf. Lang. Technol.*, 2009, pp. 1–8.

[17] K. Mehmood, D. Essam, K. Shafi, and M. K. Malik, "Sentiment analysis for a resource poor language-Roman Urdu," *ACM Trans. Asian Low-Resource Lang. Inf. Process.*, vol. 19, no. 1, pp. 1–15, 2019.

[18] W. Anwar, X. Wang, and X.-L. Wang, "A survey of automatic urdu language processing," in *Proc. Int. Conf. Mach. Learn. Cybern.*, Aug. 2006, pp. 4489–4494.

[19] G. A. Miller, "WordNet: A lexical database for English," *Commun. ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[20] 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. (ALR)*, 2009, pp. 40–46.

[21] A. Bilal, A. Rextin, A. Kakakhel, and M. Nasim, "Roman-Txt: Forms and functions of Roman Urdu texting," in *Proc. 19th Int. Conf. Hum.-Comput. Interact. Mobile Devices Services (MobileHCI)*, 2017.

[22] Y. Li and T. Yang, "Word embedding for understanding natural language: A survey," in *Guide to Big Data Applications*, vol. 26. Cham, Switzerland: Springer, 2018, pp. 83–104. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-53817-4_4

[23] T. Ahmed, "Roman to Urdu transliteration using wordlist," in *Proc. Conf. Lang. Technol.*, vol. 305, 2009, p. 309.

[24] H. Taghi-Zadeh, M. H. Sadreddini, M. H. Diyanati, and A. H. Rasekh, "A new hybrid stemming method for Persian language," *Digit. Scholarship Humanities*, vol. 32, no. 1, pp. 209–221, 2017.

[25] A. Daud, W. Khan, and D. Che, "Urdu language processing: A survey," *Artif. Intell. Rev.*, vol. 47, no. 3, pp. 279–311, Mar. 2017.

[26] H. Masroor, M. Saeed, M. Feroz, K. Ahsan, and K. Islam, "Transtech: Development of a novel translator for roman urdu to english," *Heliyon*, vol. 5, no. 5, May 2019, Art. no. e01780.

[27] F. Memood, M. Usman Ghani, M. Ali Ibrahim, R. Shehzadi, and M. Nabeel Asim, "A precisely xtreme-multi channel hybrid approach for roman urdu sentiment analysis," 2020, *arXiv:2003.05443*. [Online]. Available: http://arxiv.org/abs/2003.05443

[28] A. Z. Syed, M. Aslam, and A. M. Martinez-Enriquez, "Associating targets with SentiUnits: A step forward in sentiment analysis of urdu text," *Artif. Intell. Rev.*, vol. 41, no. 4, pp. 535–561, Apr. 2014.

[29] D. E. Kieras and M. A. Just, *New Methods in Reading Comprehension Research*. Evanston, IL, USA: Routledge, 2018.

[30] T. Ahmed and A. Hautli, "Developing a basic lexical resource for Urdu using Hindi WordNet," in *Proc. CLT*, Islamabad, Pakistan, 2010, pp. 1–8.

[31] V. Gupta, N. Joshi, and I. Mathur, "Design & development of rule based inflectional and derivational Urdu stemmer 'Usal,'" in *Proc. Int. Conf. Futuristic Trends Comput. Anal. Knowl. Manage. (ABLAZE)*, Feb. 2015, pp. 7–12.

[32] M. Alam and S. ul Hussain, "Sequence to sequence networks for roman-urdu to urdu transliteration," in *Proc. Int. Multi-topic Conf. (INMIC)*, Nov. 2017, pp. 1–7.
[33] V. Gupta, N. Joshi, and I. Mathur, "Rule based stemmer in urdu," in *Proc. 4th Int. Conf. Comput. Commun. Technol. (ICCCT)*, Sep. 2013, pp. 129–132.
[34] M. S. Husain, F. Ahamad, and S. Khalid, "A language Independent Approach to develop Urdu stemmer," in *Advances in Computing and Information Technology* (Advances in Intelligent Systems and Computing), vol. 178, N. Meghanathan, D. Nagamalai, and N. Chaki, Eds. Berlin, Germany: Springer, 2013, pp. 45–53. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-642-31600-5_5, doi: 10.1007/978-3-642-31600-5_5.
[35] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," 2014, *arXiv:1406.1078*. [Online]. Available: https://arxiv.org/abs/1406.1078
[36] O. Vinyals and Q. Le, "A neural conversational model," 2015, *arXiv:1506.05869*. [Online]. Available: https://arxiv.org/abs/1506.05869
[37] M. Z. Asghar, A. Sattar, A. Khan, A. Ali, F. Masud Kundi, and S. Ahmad, "Creating sentiment lexicon for sentiment analysis in Urdu: The case of a resource-poor language," *Expert Syst.*, vol. 36, no. 3, 2019, Art. no. e12397.
[38] T. Naseem and S. Hussain, "A novel approach for ranking spelling error corrections for urdu," *Lang. Resour. Eval.*, vol. 41, no. 2, pp. 117–128, Nov. 2007.
[39] S. Iqbal, W. Anwar, U. I. Bajwa, and Z. Rehman, "Urdu spell checking: Reverse edit distance approach," in *Proc. 4th Workshop South Southeast Asian Natural Lang. Process.*, 2013, pp. 58–65.
[40] M. Daud, R. Khan, Mohibullah, and A. Daud, "Roman urdu opinion mining system (RUOMiS)," 2015, *arXiv:1501.01386*. [Online]. Available: http://arxiv.org/abs/1501.01386
[41] S. Mukund and R. K. Srihari, "NE tagging for urdu based on bootstrap POS learning," in *Proc. 3rd Int. Workshop Cross Lingual Inf. Access Addressing Inf. Need Multilingual Societies (CLIAWS)*, 2009, pp. 61–69.
[42] M. Bilal, H. Israr, M. Shahid, and A. Khan, "Sentiment classification of Roman-Urdu opinions using Naïve Bayesian, Decision Tree and KNN classification techniques," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 28, no. 3, pp. 330–344, 2016.
[43] M. Vathsala and G. Holi, "RNN based machine translation and transliteration for Twitter data," *Int. J. Speech Technol.*, vol. 23, pp. 499–504, Jun. 2020.
[44] S. M. Ash, "Transliteration of data records for improved data matching," U.S. Patent App. 16 197 222, May 21, 2020.
[45] M. Rauf and S. Padó, "Learning trilingual dictionaries for Urdu–Roman Urdu–English," in *Proc. Workshop Widening NLP*, 2019, pp. 38–42.
[46] Z. Sharf and S. U. Rahman, "Lexical normalization of roman Urdu text," *Int. J. Comput. Sci. Netw. Secur.*, vol. 17, no. 12, pp. 213–221, 2017.
[47] M. Faruqui, P. Majumder, and S. Padó, "Soundex-based translation correction in Urdu–English cross-language information retrieval," in *Proc. 5th Int. Workshop Cross Lingual Inf. Access*, 2011, pp. 25–29.
[48] R. Aziz and M. W. Anwar, "Urdu spell checker: A scarce resource language," in *Proc. Int. Conf. Intell. Technol. Appl.* Springer, 2019, pp. 471–483.
[49] M. A. Zahid, N. I. Rao, and A. M. Siddiqui, "English to urdu transliteration: An application of soundex algorithm," in *Proc. Int. Conf. Inf. Emerg. Technol.*, Jun. 2010, pp. 1–5.
[50] N. Mathur, T. Baldwin, and T. Cohn, "Tangled up in BLEU: Reevaluating the evaluation of automatic machine translation evaluation metrics," 2020, *arXiv:2006.06264*. [Online]. Available: http://arxiv.org/abs/2006.06264
[51] M. P. Akhter, Z. Jiangbin, I. R. Naqvi, M. Abdelmajeed, and M. T. Sadiq, "Automatic detection of offensive language for urdu and roman urdu," *IEEE Access*, vol. 8, pp. 91213–91226, 2020.
[52] Y. Wu *et al.*, "Google's neural machine translation system: Bridging the gap between human and machine translation," 2016, pp. 1–23, *arXiv:1609.08144*. [Online]. Available: https://arxiv.org/abs/1609.08144
[53] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3104–3112.
[54] W. Ling, I. Trancoso, C. Dyer, and A. W. Black, "Character-based neural machine translation," 2015, *arXiv:1511.04586*. [Online]. Available: https://arxiv.org/abs/1511.04586
[55] C. D. Manning, "Effective approaches to attention-based neural machine translation," 2015, *arXiv:1508.04025*. [Online]. Available: https://arxiv.org/abs/1508.04025
[56] P. Agrawal and L. Jain, "English to Sanskrit transliteration: An effective approach to design natural language translation tool," *Int. J. Adv. Res. Comput. Sci.*, vol. 8, no. 1, pp. 103–107, 2017. [Online]. Available: http://www.ijarcs.info/index.php/Ijarcs/article/view/2860, doi: 10.26483/ijarcs.v8i1.2860.
[57] N. Jadoon Khan, W. Anwar, and N. Durrani, "Machine translation approaches and survey for indian languages," 2017, *arXiv:1701.04290*. [Online]. Available: http://arxiv.org/abs/1701.04290
[58] A. Khattak, M. Z. Asghar, A. Saeed, I. A. Hameed, S. A. Hassan, and S. Ahmad, "A survey on sentiment analysis in Urdu: A resource-poor language," *Egyptian Inform. J.*, Mar. 2020. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1110866520301171, doi: 10.1016/j.eij.2020.04.003.
[59] P. Juola, "Self-organizing machine translation: Example-driven induction of transfer functions," 1994, *arXiv:cmp-lg/9406012*. [Online]. Available: https://arxiv.org/abs/cmp-lg/9406012
[60] P. Koehn, "Europarl: A parallel corpus for statistical machine translation," in *Proc. MT Summit*, vol. 5. Citeseer, 2005, pp. 79–86.
[61] *Center of Language Engineering*. Accessed: May 5, 2020. [Online]. Available: http://www.cle.org.pk/software/localization.htm
[62] A. Hardie, "Developing a tagset for automated part-of-speech tagging in Urdu," in *Proc. Corpus Linguistics*. Lancashire, U.K.: Lancaster Univ., 2003, p. 103. [Online]. Available: https://eprints.lancs.ac.uk/id/eprint/103
[63] D. Becker and K. Riaz, "A study in Urdu corpus construction," in *Proc. 3rd Workshop Asian Lang. Resour. Int. Standardization (COLING)*, 2002. [Online]. Available: https://www.aclweb.org/anthology/W02-1201
[64] M. Gorelick and I. Ozsvald, *High Performance Python: Practical Performant Programming for Humans*. Sebastopol, VA, USA: O'Reilly Media, 2020.
[65] N. Silaparasetty, "Python programming in Jupyter notebook," in *Machine Learning Concepts with Python and the Jupyter Notebook Environment*. Berkeley, CA, USA: Springer, 2020, pp. 119–145, doi: 10.1007/978-1-4842-5967-2_7.
[66] G. Moruzzi, "Python basics and the interactive mode," in *Essential Python for the Physicist*. Cham, Switzerland: Springer, 2020, pp. 1–39, doi: 10.1007/978-3-030-45027-4_1.
[67] M. J. Denny and A. Spirling, "Text preprocessing for unsupervised learning: Why it matters, when it misleads, and what to do about it," *Political Anal.*, vol. 26, no. 2, pp. 168–189, Apr. 2018.
[68] S. Kannan, V. Gurusamy, S. Vijayarani, J. Ilamathi, M. Nithya, S. Kannan, and V. Gurusamy, "Preprocessing techniques for text mining," *Int. J. Comput. Sci. Commun. Netw.*, vol. 5, no. 1, pp. 7–16, 2015.
[69] S. Vijayarani and R. Janani, "Text mining: Open source tokenization tools-an analysis," *Adv. Comput. Intell., Int. J.*, vol. 3, no. 1, pp. 37–47, 2016.
[70] T. Hiraoka, H. Shindo, and Y. Matsumoto, "Stochastic tokenization with a language model for neural text classification," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 1–10.
[71] M. Davis and L. Collins, "Unicode," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. Conf.*, Nov. 1990, pp. 499–504.
[72] M. Needleman, "The unicode standard," *Serials Rev.*, vol. 26, no. 2, pp. 51–54, 2000.
[73] *ijunoon: Roman Urdu To Urdu Script Transliteration*. Accessed: Oct. 27, 2020. [Online]. Available: https://www.ijunoon.com/transliteration/roman-to-urdu
[74] *Google Online Translator*. Accessed: Oct. 27, 2020. [Online]. Available: https://translate.google.com/?hl=en&tab=tt
[75] M. Y. Khan and T. Ahmed, "Pseudo transfer learning by exploiting monolingual corpus: An experiment on roman Urdu transliteration," in *Proc. Int. Conf. Intell. Technol. Appl. (INTAP)*, in Communications in Computer and Information Science, I. Bajwa, T. Sibalija, and D. Jawawi, Eds. Singapore: Springer, 2019, pp. 422–431, doi: 10.1007/978-981-15-5232-8_36.
[76] S. Qazi and H. Tariq, "A novel and efficient method for roman to urdu transliteration via heuristics-based searching on parse trees," *Int. Trans. J. Eng., Manage., Appl. Sci. Technol.*, vol. 10, no. 3, pp. 567–577, 2019.

MOBEEN SHAHROZ received the M.C.S. degree from the Department of Computer Science, Khawaja Fareed University of Engineering and Information Technology (KFUEIT), Rahim Yar Khan, Pakistan, in 2018, where he is currently pursuing the M.S. degree in computer science. He is currently serving as a Research Assistant with KFUEIT. His current research interest includes the Internet of Things (IoT), artificial intelligence, data mining, natural language processing (NLP), supervised and unsupervised machine learning, and image processing.

MUHAMMAD FAHEEM MUSHTAQ received the B.S. (IT) and M.S. (CS) degrees from The Islamia University of Bahawalpur, Punjab, Pakistan, in 2011 and 2013, respectively, the Microsoft certifications of Internet Security and Acceleration (ISA) Server, Microsoft Certified Professional (MCP), Microsoft Certified Technology Professional (MCTS), in 2010, and the Ph.D. degree from the Department of Information Security, Faculty of Computer Science and Information Technology, University Tun Hussein Onn Malaysia (UTHM), Malaysia, in 2018.

He has made several contributions through research publications and book chapters toward information security. He is currently working as an Assistant Professor with the Department of Information Technology, Khwaja Fareed University of Engineering and Information Technology, Rahim yar Khan, Pakistan. He has been working as a Research Assistant during his Ph.D. degree from March 2016 to August 2018. He has been appointed as the Vice President of UTHM's Graduates Student Association from 2017 to 2018. His main research interests include information security, data mining, as well as cognitive system and applications.

ARIF MEHMOOD received the Ph.D. degree from the Department of Information and Communication Engineering, Yeungnam University, South Korea, from February 2014 to November 2017. Since November 2017, he has been an Assistant Professor with the Department Information Technology, The Islamia University of Bahawalpur, Pakistan. His current research interests include data mining, mainly working on AI and deep learning-based on text mining, and data science management technologies.

SALEEM ULLAH was born in AhmedPur East, Pakistan, in 1983. He received the B.Sc. and MIT degrees in computer science from Islamia University Bahawalpur and Bahauddin Zakariya University, Multan, in 2003 and 2005, respectively, and the Ph.D. degree from Chongqing University, China, in 2012. From 2006 to 2009, he worked as a Network/IT Administrator in different companies. From August 2012 to February 2016, he worked as an Assistant Professor with Islamia University Bahawalpur, Pakistan. He is currently working as an Associate Professor with the Khwaja Fareed University of Engineering and Information Technology, Rahim Yar Khan, since February 2016. He has almost 13 years of Industry experience in the field of IT. He is an active researcher in the field of adhoc networks, congestion control, and security.

GYU SANG CHOI (Member, IEEE) received the Ph.D. degree in computer science and engineering from Pennsylvania State University. He was a Research Staff Member with the Samsung Advanced Institute of Technology (SAIT), Samsung Electronics Company Ltd., from 2006 to 2009. Since 2009, he has been with Yeungnam University, where he is currently a Professor. His research interests include data mining, deep learning and parallel computing, while his prior research has been mainly focused on improving the performance of clusters. He is a member of ACM.