Innovative Computing Review (ICR)
Volume 2 Issue 2, Fall 2022
ISSN(P): 2791-0024 ISSN(E): 2791-0032
Homepage: https://journals.umt.edu.pk/index.php/ICR

Article QR

Title: **Saraiki Language Hybrid Stemmer Using Rule-Based and LSTM-Based Sequence-To-Sequence Model Approach**

Author (s): Mubasher H. Malik[1], Hamid Ghous[2], Iqra Ahsan[1], Maryem Ismail[1]

Affiliation (s): [1]Department of Computer Science, Institute of Southern Punjab Multan, Pakistan
[2]Australian Scientific & amp; Engineering Solutions, Sydney, New South Wales, Australia

DOI: https://doi.org/10.32350.icr.22.02

History: Received: October 10, 2022, Revised: November 11, 2022, Accepted: December 2, 2022

Citation: M. H. Malik, H. Ghous, I. Ahsan, and M. Ismail, "Saraiki language hybrid stemmer using rule-based and LSTM-Based sequence-to-sequence model approach," *UMT Artif. Intell. Rev.,* vol. 2, no. 2, pp. 17-40, 2022, doi: https://doi.org/10.32350.icr.22.02

Conflict of Interest: Author(s) declared no conflict of interest

A publication of
School of Systems and Technology
University of Management and Technology, Lahore, Pakistan

Saraiki Language Hybrid Stemmer Using Rule-Based and LSTM-Based Sequence-To-Sequence Model Approach

Mubasher H. Malik[1*], Hamid Ghous[2], Iqra Ahsan[1], and Maryem Ismail[1]

[1]Department of Computer Science, Institute of Southern Punjab, Multan, Pakistan
[2]Australian Scientific & Engineering Solutions, Sydney, New South Wales, Australia

Abstract- **Converting a word to its original form, is called stemming, which is extremely important in the field of Natural language processing (NLP). It's an integral part of the linguistic pre-processing of every Natural language processing application. Stemming converts inflectional word forms into their root word. Much work has been done for stemming in different national and regional languages like English, French, Arabic, German, Urdu, and Hindi. Many regional languages still need work to build digital resources using Natural language processing. Saraiki is one of the widely spoken regional languages in Pakistan. Almost eighty million people use this language for communication. There are very limited digital resources using the Saraiki language available to support advancement in Natural language processing technologies. The current research aims to propose a hybrid stemmer to stem Saraiki Work. The hybrid stemmer contains two hundred prefix and postfix rules and Long short-term memory based sequence-to-sequence model for converting Saraiki words into the stem. Firstly, Saraiki text was pre-processed, and a rule set was implemented. Secondly, the Long short-term memory based sequence-to-sequence model was deployed to stem the Saraiki word correctly. In the last step, The Saraiki Stemmer performance was evaluated by accurately finding stem word accuracy using a rule-set and Long short-term memory sequence to sequence model. After experiments, using the rule set correctly, stem word accuracy was 68.53%, while the Long short-term memory based sequence-to-sequence model produced 93.0% accuracy of correctly stem words. This work contributes significantly to the regional linguistic field by introducing stemmer for the Saraiki language.**

Index Terms-**Hybrid Stemmer, LSTM, Rule-based Stemmer, Saraiki, Stemming**

I. Introduction

The current research is categorized into sub-sections. Section 1 depicts the introduction of the Saraiki language, its history, origin, and varieties. It includes the

[*] Corresponding Author: mubasher@isp.edu.pk

role of Natural language processing (NLP) research for low-resource languages such as Saraiki Language. This section also includes Sarariki language alphabets, facts about stemming, and its types.

A. Overview of Saraiki Language

Saraiki is an Indo-Aryan language widely used in Pakistan and India [1]. Saraiki is the only language of eighty million people in Pakistan, ranging across all four provinces of Pakistan, with a majority of speakers in southern Punjab [2]. Saraiki originated from the word "Sauvira," an ancient kingdom also mentioned in the Sanskrit epic Mahabharata [3].

Saraiki language is considered as a dialect of Punjabi by most British Colonial Administrators [4].There are several varieties of the Saraiki language, such as central Saraiki, which is most famous and spoken in Multan, Muzaffargarh, Bahawalpur, and Dera Ghazi Khan, Pakistan [5] . The other varieties are southern Saraiki, Sindhi Saraiki, Kulachlwall Saraiki, Northern Saraiki, and Eastern Saraiki. All these varieties have minor regional variations in punctuation [6], [7].

Natural Language Process (NLP) employs computational techniques to learn, understand, and produce human language content [8]. Early computational approaches to language research focused on automating the analysis of the linguistic structure of language and developing basic technologies such as machine translation, speech recognition, and speech synthesis [9], [10]. Today's researchers refine and use such tools in real-world applications, creating spoken dialogue systems, and speech-to-speech translation engines, mining social media for information about health or finance, identifying sentiment, emotion toward products, and services [11].

The major challenge for researchers in the field of NLP is low-resource languages such as the Saraiki language. NLP research may focus on creating novel language resources and benchmarks; some may customize existing NLP solutions to new languages and domains [12]. In parallel, there can be research that actively explores new NLP techniques that could generalize to different low-resource setups - in terms of data availability and the availability of computational resources [13]. The prime purpose of this paper is to develop a Saraiki text stemmer that would eventually open a door for further analysis of the texts written in the Saraiki language.

B. Saraiki Language Alphabets

There are around eighty million native language users in Pakistan and India only. It is written in Perso-Arabic script; however, it has its own set of alphabets that consists of 45 letters. Of these forty-five letters, thirty-nine are the same as the Urdu language, and six are additional letters. However, some researchers consider it as a dialect of the standard Punjabi language; whereas, it is a separate language with its own identity [14], [15]. Table I shows the alphabet of the Saraiki language and its variation of sounds in English.

Table I
Saraiki Language Alphabets and Sounds

Saraiki Alphabet	English Sound	Saraiki Alphabet	English Sound	Saraiki Alphabet	English Sound
ا	a	ڑ	ṛ	گ	g
ب	b	ز	z	ل	l
پ	p	ژ	zh	م	m
ت	t	س	s	ن	n
ٹ	ṭ	ش	sh	و	v, o, or ū
ث	th	ص	ṣ	ہ, ـہـ, ـہ	h
ج	j	ض	z̤	ھ	h
چ	ch	ط	t	ء	'
ح	h	ظ	ẓ	ی	y, i
خ	kh	ع	'	ے	ai or e
د	d	غ	gh		
ڈ	ḍ	ف	f		
ذ	dh	ق	q		
ر	r	ک	k		

C. Stemming and Types

Stemming is the linguistic process of converting inflected forms of words to their basic form [16]–[18]. Stemming is a challenging task for a large number of ambiguous structure languages and poor resource languages [19], [20]. Stemming has been seen from various perspectives by academics working in information retrieval (IR) [21]. Stemming can improve the performance of text processing because this process would merge words that have the same root. If words have the same roots, they are considered to have a semblance of meaning. Therefore, documents with words with the same roots are considered relevant so that the steaming process would reduce the features dimension of the documents [22]. It's a linguistic procedure in which the numerous morphological forms of a word are mapped to a single word form, known as the infinitive form [23]. A stemmer, for example, maps the words: maintained, maintaining, and maintenance from the root word maintain'. Similarly, in the Saraiki language: Words like تے کھیڈا کھیڈن، کھیڈے can be reduced to the root word کھیڈ.

Stemming algorithms have been developed during the last few decades to automate the process which requires different approaches as it is mostly language-specific [Stemming algorithms were classified into statistical stemmers and Rule-based stemmers. Statistical stemmers, also known as unsupervised stemmers, use training data for performing stemming. Rule-based stemmers use a set of rules defined to perform stemming [24], [26].

The statistical stemmers are inaccurate and fail to take advantage of some language phenomena that simple rules can easily express. On the other hand, handcrafting the stemming rules in the rule-based stemmers is a time-consuming, tedious, and impractical task [27]. In this paper, a hybrid stemmer was proposed to develop a linguistics resource for the poor language Saraiki.

In the next section, different researchers' contributions to the development of stemming algorithms are presented in detail.

II. Literature Review

Stemmers play an important role in improving the performance and efficiency of any IR system. Researchers have contributed a lot to empower this area of linguistics for a better understanding of different languages by using modern tools and techniques. The current section describes the contribution of researchers in stemming algorithms proposed for various national and international languages such as English, Arabic, Persian, Hindi, Urdu, Sindhi, Hindko, and many others. The review of literature is divided into three sub-sections: Rule-based stemming, statistical stemming, and hybrid stemming.

A. Rule-based Stemmers

A Marathi Language stemmer was proposed in 2022. Supervised ML techniques adopted. Handwritten grammar rules and word dictionary of Marathi language was developed. The proposed system produced 61.36% accuracy [28]. A Telegu language stemmer was proposed in 2022. Later rules were developed to analyze Telegu text [29]. A Sindhi Language stemmer was proposed in 2021, and Lexicon-based affix removal was deployed through stemming algorithms. Thirty-Eight linguistics rules and fifty thousand words lexicons were adopted for the steaming process. The stemmer produced an accuracy of 84.85% [30]. A Punjabi Language rule-based stemmer was proposed in 2020. Brute-force and suffix-stripping techniques were deployed. A lexicon containing 1762 root words was taken as an input. The stemmer stems 1564 words properly produced 88.75% accuracy [31]. A Bengali language rule-based stemmer was proposed in 2020. WordNet was used to produce better outcomes. A Bengali corpus containing 500 sentences was adopted. The stemmer produced 98.86% accuracy for nouns and 99.75% for verbs [32].

A Sinhalese language rule-based stemmer was proposed in 2020 for employing suffix and prefix rules. Stemming effectiveness was evaluated using Naïve Bayes, Random Forest, and Support Vector Machine classifiers. Two datasets containing words related to cricket, rugby, football, athletics, and academics were used. The stemmer produced promising results [31],[33]. An Assamese language rule-based stemmer was proposed in 2019 using the suffix stripping method, and WordNet was adopted for stemming. Rules were developed for nouns and verbs. The stemmer achieved 85% accuracy [34]. A Sundanese language rule-based stemmer was proposed in 2018. A dataset of 4,453 Sundanese unique attached words was accessed from 70 Balebat articles and 68 Dewan Dakwah Jabar articles for the experimental study. The stem and affix sequences are manually identified in the dataset. The results showed that stemmer exceeds the modified baseline regarding correctly stemmed affixed words and identified attached type accuracy. Stemmer successfully affixed 68.87% of Sundanese affixed types and created 96.79% of correctly affixed words [35].

An Urdu language rule-based stemmer was proposed in 2017. Six infix word classes were created. Four corpora were used to create a directory of words. These corpora contain words related to Urdu news headlines, politics, weather, sports, Urdu dictionaries, and grammar books. The stemmer achieved 87.4% accuracy [36]. A Ge'ez Language stemmer was introduced in 2017. An affix removal approach was deployed. The stemmer was based on evaluating over-stemming, under-stemming, and structured laws. Manual error counting was used for evaluation. The stemmer achieved 75.95% accuracy [36]. Another Gujarati language rule-based stemmer was proposed in 2014. Rules were defined for Gujarati text EMILLE corpus was used to evaluate the stemmer. The stemmer achieved 92.41% accuracy [37]. A Nepali language rule-based stemmer was proposed in 2014. Affix stripping techniques were adopted for stem extraction from the Nepali text. A corpus of 1800 words was used for the evaluation of the stemmer. The stemmer produced 90.48% accuracy [38], [39].

B. Statistical Stemmers

The employment of statistical tools is another prevalent option. In such techniques, the system learns from the existing corpus. It makes decisions about unknown material based on knowledge gained through experience, for instance, statistical measurements learned through experience are utilized to eliminate prefixes and affixes. However, statistical approaches are widely used in the language processing community. N-gram-based statistics are used by W. B. Frakes [40]. Melucci [41] used HMMs and YASS: Yet another suffix stripper by Majumder [42] as a few examples. Such techniques are restricted in terms of gaining experience. In other words, such techniques seldom cover the language's whole grammar [43].The N-gram technique, for example, is superior at removing only suffixes. Still, the Urdu language contains suffixes, co-suffixes, prefixes, infixes, and circumfixes (prefixes and suffixes simultaneously) [44], and the use of HMM requires that every word must start with the prefix [41]. This review demonstrates that statistical techniques for stemming are insufficient. The book contains a complete study of stemming techniques used in Urdu, Persian, and Arabic [45].

A Gujarati Language statistical stemmer was proposed in 2021. A supervised machine learning algorithm was adopted to assess the accuracy of the web page classification. The stemmer eliminates the inflectional or derivational form of words to its root stem. The stemmer produced 97% accuracy [46]. A Marathi Language statistical stemmer was proposed in 2021, namely MT stemmer. The stemmer focused on removing suffixes to retrieve the root word form gender-based suffixes and auxiliary verb-based suffixes namely two-stage stemming were performed by stemming. The average precision, recall, and F-1 score was 0.706, 0.806, and 0.753, respectively [47]. An Arabic language statistical stemmer was introduced in 2021. The major focus was the infix patterns of Arabic word lists. TREC2002, a standard dataset, was used for text retrieval. The performance of the stemmer was evaluated using three stemmers Condlight, Light10, and ARLS. The proposed Arabic stemmer produced promising results with precision, recall, F-measure, and ICF at 60%, 79%, 68%, and 81%, respectively [48].

C. Hybrid Stemmers

A hybrid stemmer combines the rule base stemmer and statistical stemmer [49]. A hybrid Urdu language stemmer was proposed in 2020. Word and text corpus was used as dictionaries. Affix-stemming rules were deployed. The stemmer produced promising results [31] [33]. A Gujarati hybrid stemmer was proposed in 2017 for removing affixes, from Gujarati words and get optional results. The Gujarati stemmer searches an online Gujarati dictionary for the stem word. EMILE corpus was used for stemming. The stemmer produced 97.09% accuracy [50]. A Punjabi language hybrid stemmer was proposed in 2017. A dataset containing 2.5 million tokens was used. A rule set contains sixty-three affixes rules created. The stemmer achieved 86.01% accuracy [51].

A hybrid stemmer for the Persian language was introduced in 2015. The affix removal method was adopted. Intervening and Makassar wordlists of Persian words were taken. The stemmer produced 97% accuracy [52]. An Arabic language hybrid stemmer was proposed in 2013. An Arabic language dictionary containing Nine thousand words was used. Prefix and suffix rules were deployed. The stemmer produced promising results [53]. A Gujarati hybrid stemmer was proposed in 2011. A suffix stripping technique was adopted. Part of the module was also used. The stemmer achieved promising results [54].

Hence, it is concluded that there exist many resource-poor languages lacking linguistics research Saraiki is one such language, in which limited research has been conducted. So, there is a dire need to develop resources for Saraiki Language to conduct a proper research study on texts written in Saraiki language. Therefore, a hybrid Saraiki language stemmer is proposed in the current research. The next section shows the proposed methodology for Saraiki Language hybrid stemmer.

III. Materials and Methods

This section contains the proposed methodology for a hybrid Saraiki language stemmer. This section includes the proposed algorithm, corpus creation, rules creation, and sequence-to-sequence model.

The model adopted for this work is shown in figure Firstly, preprocessing techniques such as tokenization, space removal, punctuation removal, and digital removal are applied to Saraiki Corpus. Secondly, preprocessed data was processed through the Rule-based stemming module and LSTM sequence to sequence model for generating stem words. Finally, the results were compared for the two techniques.

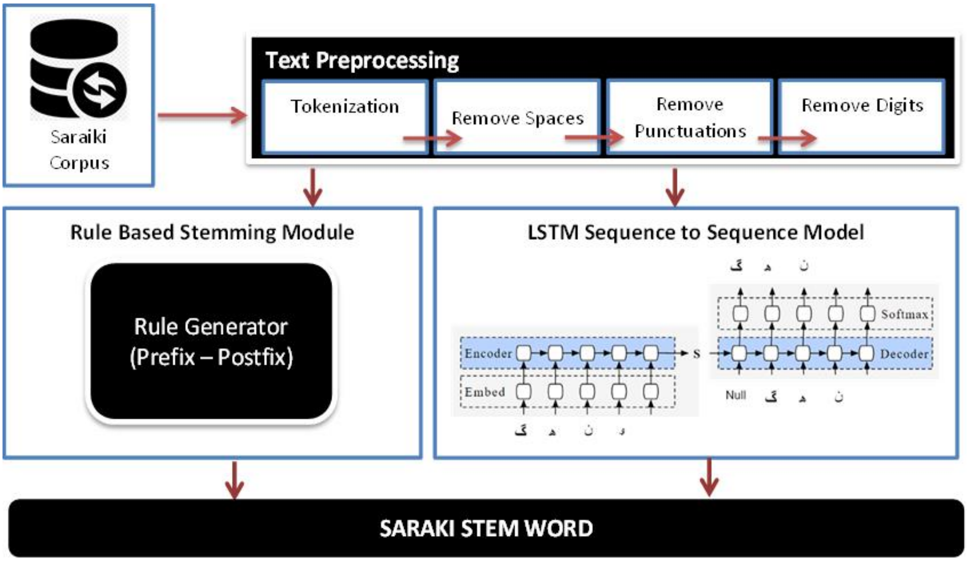


Fig. 1. Proposed model for Saraiki language hybrid stemmer

A. Saraiki Language Corpus

A large amount of text was required to build the Saraiki language corpus. It was not easy to collect Saraiki language text as there is no such publicly available dataset of Saraiki language for stemming purposes. The dataset was manually annotated by creating prefix and postfix word lists. The dataset contains 10000 Saraiki words taken from Saraiki dictionary and documents available on a different website. Table II shows the Saraiki language dataset words.

Table II
Saraiki Language Dataset

Saraiki Words	Prefix	Postfix
لکھدِن	لکھ	دِن
منزلاں	منزل	اں
کڈھݨ	کڈھ	ݨ
کھیݙݨ	کھیݙ	ݨ
بھجݨا	بھج	ݨا
ونڄیں	ونڄ	یں
پڑھدا	پڑھ	دا
کھاوݨ	کھا	وݨ
لکھسی	لکھ	سی
گھنو	گھن	و
کریندے	کر	یندے
گھنو	گھن	و
ݙساؤ	ݙس	اؤ
ملدے	مل	دے
بہشتی	بہشت	ی
خطاط	خط	اط
ٻاہرلی	ٻاہر	لی
ملتانی	ملتان	ی
ݙیکھدا	ݙیکھ	دا
آکھیا	آکھ	یا
ملیا	مل	یا
اوازیں	اواز	یں
کتاباں	کتاب	اں
چھاپی	چھاپ	ی
قربانیاں	قربان	یاں
لبھدے	لبھ	دے

The taken was pre-processed before the creation of Saraiki words lists. Word tokens were created, and punctuations, special characters, extra spaces, and digits were removed. Finally, the Saraiki language corpus was developed for performing the stemming linguistic process.

B. Rule-based Stemming Module

Saraiki language hybrid stemmer is based on two modules. Firstly, the rule-based stemmer module. This module deploys prefix and postfix rules on Saraiki corpus to stem Saraiki words. Two hundred fifty postfix and prefix rules were created. A minimum word length rule was also created. After the word has been pre-processed, prefix and postfix rules are deployed. After the applicable rule's determination, the word is broken down, and a list of possible words is generated. If relevant rules unable to determine, the same term as the root word is returned. To discover frequencies, the possibilities list is compared with the dataset. The word with the highest frequency is returned as a root or stem word. The most often occurred word has the highest possibility of being the root or stem word. The following are some of the postfix and prefix rules.

C. Minimum Word Length Rule

After a detailed analysis of Saraiki morphology, it is observed that a Saraiki word comprised of only two or three characters which is already a stem word. For example, the word توں (You), وڈٖا (Elder) are already stem words. These words are treated as stem words and filtered out to avoid further stemming processing. The finding of this rule is a novel contribution of the proposed Saraiki stemmer. Some example words identifying this rule are given in Table III.

Table III

Example of Words Handles by minimum Word Length Rule

فن	توں	تے	رسم
گھن	ڈٖلھ	وطن	جاہ
وں	مار	مٹی	مُلھ
وچ	ابّا	وڈٖے	وچ

D. Postfix and Prefix Rules

The following showed some postfix and prefix rules for stemming Saraiki words. Two hundred and fifty rules deployed using hybrid Saraiki language stemmer.

Rule no 1: If the token has ending characters دِن then remove دِن from the end. For example لکھدِن to لکھ.

Rule no 2: If the token has ending characters اں, then remove اں from the end. For example, منزلاں to منزل.

Rule no 3: If the token has ending characters ݨ, remove ݨ from the end. For example کڈھݨ to کڈھ.

Rule no 4: If the token has ending characters ,ٹا then remove ٹا from the end. For example بھجٹا to بھج.

Rule no 5: If the token has ending characters یں then remove یں from the end. For instance, ونجیں to ونج.

Rule no 6: If the token has finalizing characters دا then remove دا from the end. For example, پڑھدا to پڑھ.

Rule no 7: If the token has ending characters وݨ, then remove وݨ from the end. For example کھاوݨ to کھا.

Rule no 8: If the token has ending characters سی then remove سی from the end. For example لکھسی to لکھ.

Rule no 9: If the token has ending characters و, remove و from the end. For instance, گھنو to گھن.

Rule no 10: If the token has ending characters یندے then remove یندے from the end. For example, کریندے to کر.

Rule no 11: If the ticket has ending characters ,اؤ then remove اؤ from the end. For example, ڈٖساؤ to ڈٖس.

Rule no 12: If the token has ending characters دے then remove دے from the end. For example, ملدے to مل.

Rule no 13: If the token has ending characters ی, remove ی from the end. For example بہشتی to بہشت.

Rule no 14: If the token has ending characters اط then remove اط from the end. For example خطاط to خط.

Rule no 15: If the token has ending characters لی then remove لی from the end. For example ٻاہرلی to ٻاہر .

Rule no 16: If the token has ending characters یا then remove یا from the end. For example آکھیا to آکھ.

Rule no 17: If the token has ending characters یا then remove یا from the end. For example ملیا to مل.

Rule no 18: If the token has ending characters یاں then remove یاں from the end. For example, قربانیاں to قربان.

Rule no 19: If the token has ending characters ی, remove ی from the end. For example تعریفی to تعریف.

Rule no 20: If the token has ending characters یندن then remove یندن from the end. For example اَکھیندن to اَکھ.

Rule no 21: If a word starts with بد) bay+daal) then remove بد) bay+daal) from beginning. صورت – بدصورت For example (badsūrat) (sūrat)

Rule no 22: If a word starts with بے) bay+badi- ye), then remove بے) bay+badi-ye) from beginning - کدر بکدر For example (bēkdar) (kadar)

E. LSTM-based Sequence to Sequence Model

Sequence to sequence is the most efficient approach for automatically converting the script of a word from a source sequence to a target sequence. Sequence to sequence modeling is one of the intriguing applications of NLP. Long Short-term Memory (LSTM), which is a special kind of RNN (Recurrent Neural Network) [55]. LSTM networks are suitable for analyzing sequences of text data and to predict the next word. LSTM could be a good solution if you want to indicate the next point of a given time sequence [56].

In this work, LSTM based sequence to sequence model is deployed for stemming Saraiki words. The language translation example is shown in Figure 2.

F. Character-based Sequence to Sequence LSTM

Previous researches have often used a pre-processing phase to extract words from source sequences and develop a vast vocabulary that aids in the one-hot transformation of an input sequence into fixed-length vectors. The chosen language, however, cannot include all of the terms in the data set because of the sparsity and high dimension of this word-level representation. As a result, special tokens are frequently used to substitute OOV terms. To achieve the final translation results, a post-processing step is necessary to manage the UNKs in the output sequence. In this research, we describe the LSTM encoder and decoder, a technique for character-level sequence encoding using RNN, as given in Figure 3.

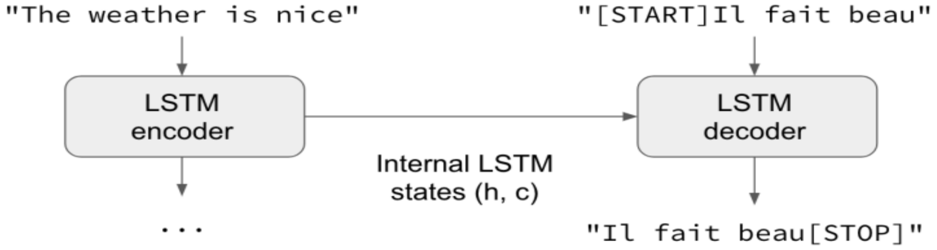


Fig. 2. LSTM sequence to sequence model [57]

F. Character-based Sequence to Sequence LSTM

Previous researches have often used a pre-processing phase to extract words from source sequences and develop a vast vocabulary that aids in the one-hot transformation of an input sequence into fixed-length vectors. The chosen language, however, cannot include all of the terms in the data set because of the sparsity and high dimension of this word-level

representation. As a result, special tokens are frequently used to substitute OOV terms. To achieve the final translation results, a post-processing step is necessary to manage the UNKs in the output sequence. In this research, we describe the LSTM encoder and decoder, a technique for character-level sequence encoding using RNN, as given in Figure 3.

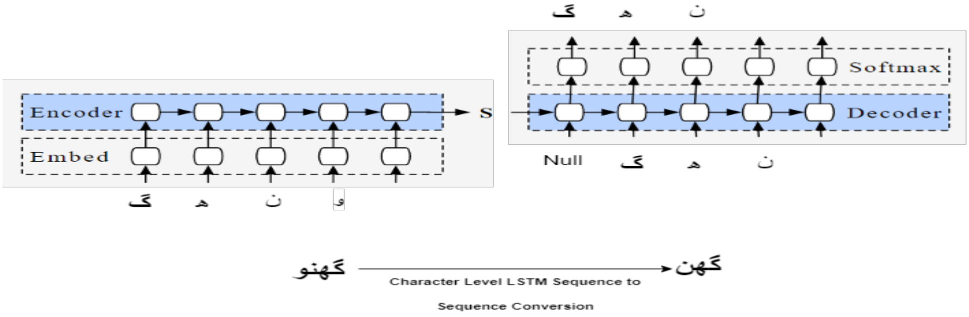


Fig.3. LSTM Saraiki stemmer

IV. Results and Discussion

This section discusses the performed experiments using a rule-based stemmer module and LSTM-based sequence to sequence Model. The Saraiki language dataset was taken for experiments, and results were discussed by analyzing the performance of the hybrid Saraiki stemmer.

A. Experiment 1 – Rule-based Stemmer Module

In this experiment, prefix and postfix rules are deployed on the Saraiki language dataset. Firstly, the rules were extracted and then applied to the given words. The rule-based technique gave the following results. Ten thousand words from the Saraiki language dataset were taken, and two hundred fifty prefix-postfix rules were applied to these words. After experiments, it was observed that the rule-based stemmer module correctly stems six thousand eight hundred and thirty-five words, while other words were wrongly stemmed. The overall accuracy of the rule-based stemmer was observed at 68.53%.

The graphical representation of correctly and wrongly stem words is shown in Figure 4.

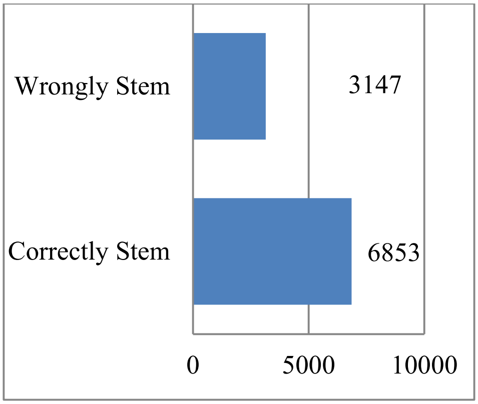


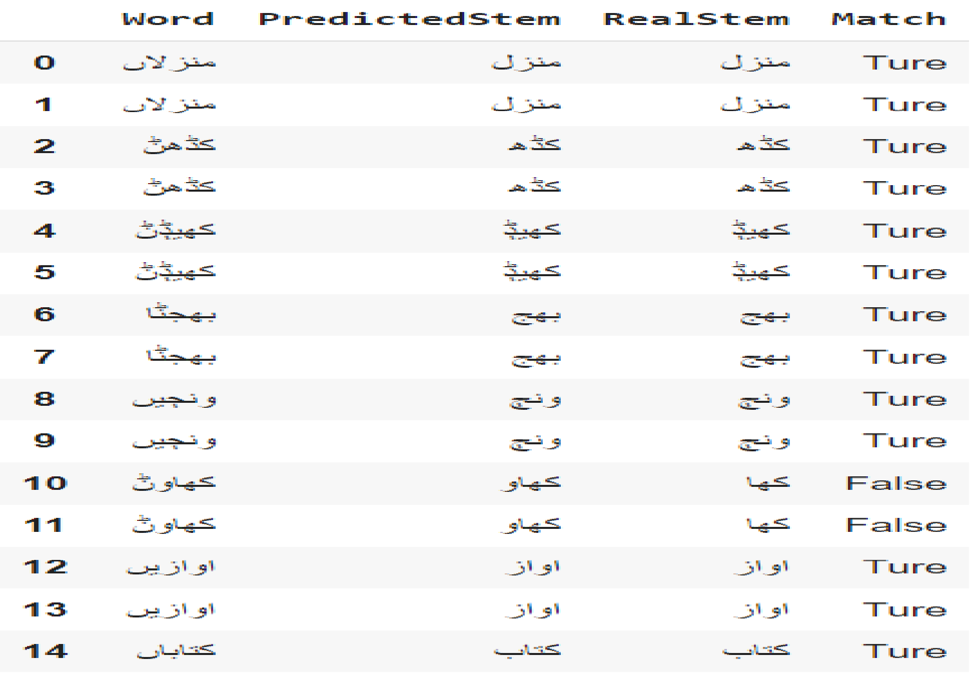
Fig.4. Rule-based stemmer - correctly and wrongly stem words

Figure 5 shows the actual Saraiki word and predicted stem word by rule-based stemmer and real stem word. Experimental results are also showed in Figure 5 as true or false matches.

Similarly, Figure 6 showed wrongly predicted Saraiki stem words by rule-based stemmer module. Finally, after deploying a rule-based stemmer module using the Saraiki dataset of 10000 words, 68.53% accuracy was achieved by the stemmer.

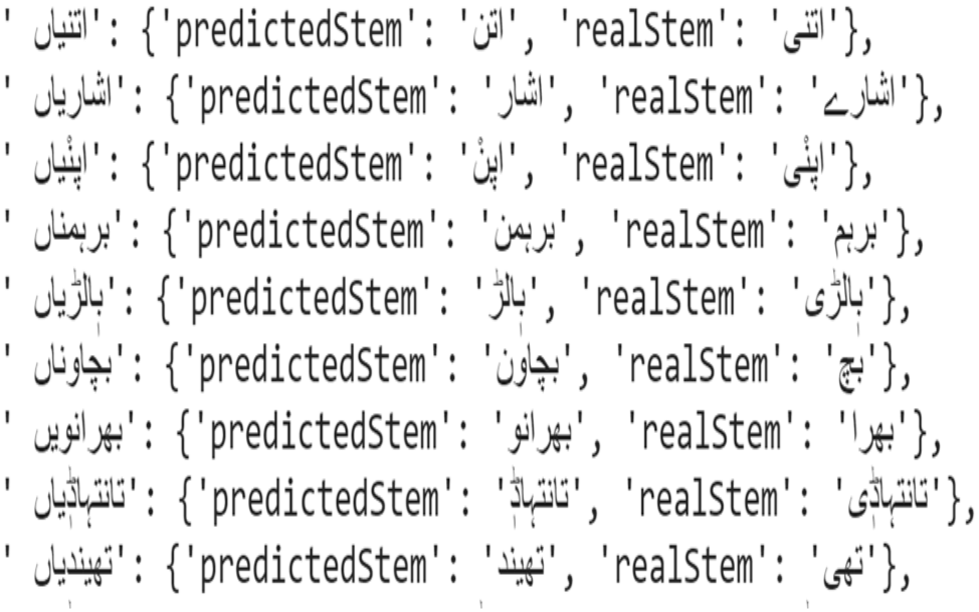
B. Experiment 2 – LSTM Based Sequence to Sequence Model

In this experiment, LSTM based sequence to sequence model was deployed using the Saraiki dataset to predict the stem of the Saraiki words. The following parameters were set before deploying LSTM based sequence to sequence model.



	Word	PredictedStem	RealStem	Match
0	منزلاں	منزل	منزل	Ture
1	منزلاں	منزل	منزل	Ture
2	کڈھݨ	کڈھ	کڈھ	Ture
3	کڈھݨ	کڈھ	کڈھ	Ture
4	کھیڈݨ	کھیڈ	کھیڈ	Ture
5	کھیڈݨ	کھیڈ	کھیڈ	Ture
6	بھجݨا	بھج	بھج	Ture
7	بھجݨا	بھج	بھج	Ture
8	ونجیں	ونج	ونج	Ture
9	ونجیں	ونج	ونج	Ture
10	کھاوݨ	کھاو	کھا	False
11	کھاوݨ	کھاو	کھا	False
12	اوازیں	اواز	اواز	Ture
13	اوازیں	اواز	اواز	Ture
14	کتاباں	کتاب	کتاب	Ture

Fig. 5. Rule-based stemmer results



```
' اتنیاں': {'predictedStem': 'اتن', 'realStem': 'اتنی'},
' اشاریاں': {'predictedStem': 'اشار', 'realStem': 'اشارے'},
' اپنْیاں': {'predictedStem': 'اپنْ', 'realStem': 'اپنْی'},
' برہمناں': {'predictedStem': 'برہمن', 'realStem': 'برہم'},
' بالڑیاں': {'predictedStem': 'بالڑ', 'realStem': 'بالڑی'},
' بچاوناں': {'predictedStem': 'بچاون', 'realStem': 'بچ'},
' بھرانویں': {'predictedStem': 'بھرانو', 'realStem': 'بھرا'},
' تانتہاڈیاں': {'predictedStem': 'تانتہاڈ', 'realStem': 'تانتہاڈی'},
' تھیندیاں': {'predictedStem': 'تھیند', 'realStem': 'تھی'},
```

Fig. 6. Rule-based stemmer - wrongly predicted words

# Batch size for training.

batch_size = 64

# Number of epochs to train for.

epochs =100

# Latent dimensionality of the encoding space.

latent_dim =256

# Number of samples to train on.

num_samples = 10000

After experiments, it was observed that LSTM based sequence to sequence model achieved 99% accuracy, while validation accuracy was 96%.

Figure 7(a) shows the training and validation accuracy comparison, while Figure 7(b) shows the training and validation loss.

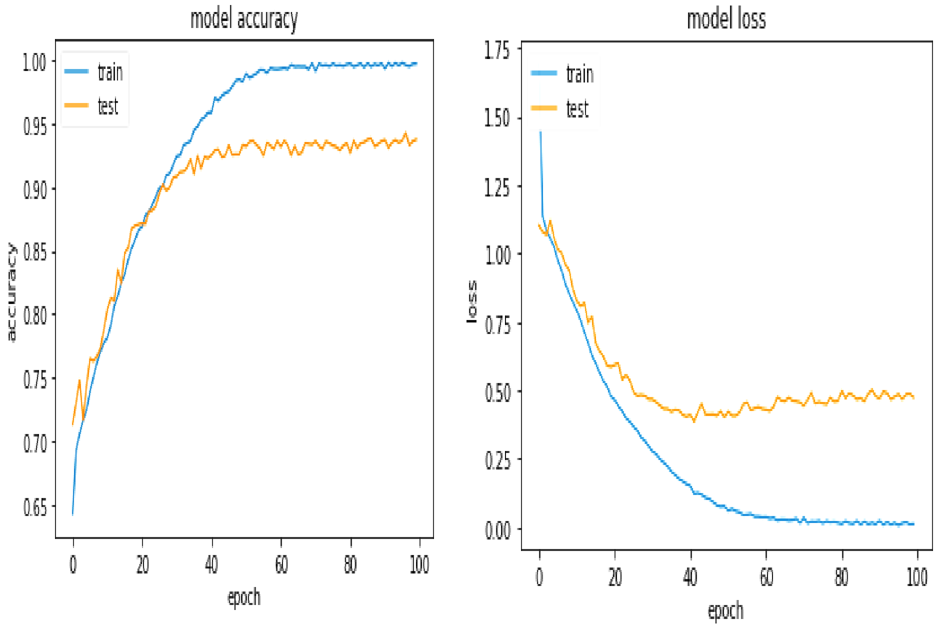


Fig. 7: (a) Comparison of training and validation accuracy (b) loss of LSTM sequence to sequence model

Here are some stem words generated by LSTM sequence to sequence model stemmer.

Therefore, the rule-based stemmer produced 68.53% accuracy after experiments, while LSTM based sequence-sequence model produced 99% accuracy, as shown in Figure 8.

Table IV
Stem Words Generated by LSTM Sequence to Sequence Model Stemmer

Input sentence: لکھدِن	Input sentence: کڈھݨ
Decoded sentence: لکھ	Decoded sentence: کڈھ
Input sentence: منزلاں	Input sentence: کھیݙݨ
Decoded sentence: منزل	Decoded sentence: کھیݙ
Input sentence: بھجݨا	Input sentence: گھنو
Decoded sentence: بھج	Decoded sentence: گھن
Input sentence: ونچیں	Input sentence: کریندے
Decoded sentence: ونج	Decoded sentence: کر
Input sentence: پڑھدا	Input sentence: گھنو
Decoded sentence: پڑھ	Decoded sentence: گھن
Input sentence: کھاوݨ	Input sentence: ݙساؤ
Decoded sentence: کھا	Decoded sentence: ݙس
Input sentence: لکھسی	Input sentence: ملدے
Decoded sentence: لکھ	Decoded sentence: مل

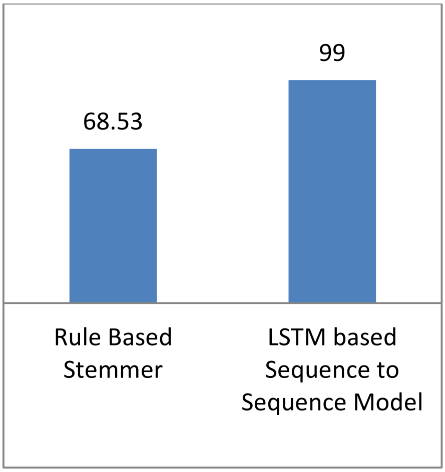


Fig. 8. Hybrid Saraiki stemmer (rule-based and LSTM-based sequence to sequence model

A. Conclusion

Stemming is a strategy used in IR systems to improve retrieval accuracy by addressing the problem of document and vocabulary mismatch during indexing and searching. The model presented in this work provides a stemmer application for the Saraiki language. The stemmer was evaluated using two approaches, namely rule-based and LSTM sequence to sequence. In our experiment, the rules-based approach gave 68% accuracy, while LSTM gave 93% accuracy. Hence, LSTM sequence to sequence is the best method to provide accurate stemmer for the Saraiki language. Future research can enhance the result by increasing the size of the dataset and mathematical modification of the classifier.

B. Future Work

In the future, the current research can be extended by extending the stemmer's ability and adding more type of rules. This can lead to better sentiment analysis and opinion mining using texts written in Saraiki language.

References

[1] E. Bashir and T. J. Conners "A descriptive grammar of Hindko, Panjabi, and Saraiki," in *A Descriptive Grammar of Hindko, Panjabi, and Saraiki*, De Gruyter Mouton, 2019.

[2] Z. L. Atta, "Saraiki," *J. Int. Phonetic Association,* pp. 1–21, 2020.

[3] A. H. Dani, "Sindhu-Sauvira: A glimpse into the early history of Sind," in *Sind Through the Centuries*, Karachi: Oxford University Press, 1981, pp. 35–42.

[4] T. Rahman, "*Language and politics in Pakistan,*" Oxford University Press, 1996.

[5] R. S. Hashmi and G. Majeed, "Saraiki ethnic identity: Genesis of conflict with state," *J Poli. Stud.*, vol. 21, no. 1, pp. 79–101, 2014.

[6] M. A. Wagha, "The development of Siraiki language in Pakistan," Ph.D. dessertation, Sch. Orient. African Stud., Univ. London, UK, 1997.

[7] C. Shackle, "The Siraiki language of central Pakistan: A reference grammar," Sch. Orient. African Stud., Univ. of London, 1976.

[8] J. Hirschberg and C. D. Manning, "Advances in natural language processing," *Science*, vol. 349, no. 6245, pp. 261–266, Jul. 2015.

[9] J. H. Paik and S. K. Parui, "A fast corpus-based stemme," *ACM Transac. Asian Lang. Inform. Proce.*, vol. 10, no. 2, pp. 1–16, June 2011.

[10] C. Parsing, "Speech and language processing," *Power Point Slides*, 2009,

[11] D. Khurana, A. Koli, K. Khattar, and S. Singh, "Natural language processing: State of the art, current trends and challenges," *Multimed. Tools Applic.*, pp. 1–32, July 2022, doi:

https://doi.org/10.1007/s11042-022-13428-4

[12] B. P. King, "Practical Natural Language Processing for Low-Resource Languages," Ph.D. thesis, Uni., Michigan, 2015.

[13] M. A. Hedderich, L. Lange, H. Adel, J. Strötgen, and D. Klakow, "A survey on recent approaches for natural language processing in low-resource scenarios," *arXiv preprint arXiv:2010.12309*, 2020,

[14] S. S. Hussain, "The growth of Saraiki language," *Pakistan J. Soc. Sci.*, vol. 36, no. 1, pp. 387–396, 2021.

[15] G. Raza, "Reduction of compound adpositions in Persian, Urdu and Saraiki," in *6th Int. Contras. Lingu. Conf., Berlin*, 2010.

[16] D. Bijal and S. Sanket, "Overview of stemming algorithms for Indian and Non-Indian languages," *arXiv preprint arXiv:1404.2878*, 2014, doi: https://doi.org/10.48550/arXiv.1404.2878

[17] B. A. Pande and H. S. Dhami, "Application of natural language processing tools in stemming," *Int. J. Comput. Applic.,* vol. 27, no. 6, pp. 14–19, 2011.

[18] D. Khyani, B. S. Siddhartha, N. M. Niveditha, and B. M. Divya, "An Interpretation of Lemmatization and Stemming in Natural Language Processing," *J. Univ. Shanghai Sci. Technol.*, vol. 22, no. 10, pp. 350–357, 2021.

[19] S. Jusoh, "A study on nlp applications and ambiguity problems," *J. Theoret. Appl. Inform. Technol.*, vol. 96, no. 6, Mar. 2018.

[20] P. Deshpande and S. Jahirabadkar, "A survey on statistical approaches for abstractive summarization of low resource language documents," in *Smart Trend Comput. Commun.*, Springer, 2022, pp. 729–738, doi: https://doi.org/10.1007/978-981-16-4016-2_69

[21] C. Moral, A. de Antonio, R. Imbert, and J. Ramírez, "A survey of stemming algorithms in information retrieval," *Inform. Res.*, vol. 19, no. 1, Mar. 2014.

[22] A. S. Rizki, A. Tjahyanto, and R. Trialih, "Comparison of stemming algorithms on Indonesian text processing," *TELKOMNIKA*, vol. 17, no. 1, pp. 95–102, 2019.

[23] S. R. Payne, J. Kodner, and C. Yang, "Learning Morphological Productivity as Meaning-Form Mappings," *Proc. Soc. Comput. Ling.*, vol. 4, no. 1, pp. 177–187, 2021, doi: https://doi.org/10.7275/rbhm-c353

[24] K. Swain and A. K. Nayak, "A review on rule-based and hybrid stemming techniques," *Int. Conf. Data Sci. Business Anal.*, IEEE, Sep. 21–23, 2018, pp. 25–29, doi: https://doi.org/10.1109/ICDSBA.2018.00012

[25] B. Gobin-Rahimbux, I. Maudhoo, and N. Gooda Sahib, "KreolStem: A hybrid language-dependent stemmer for Kreol Morisien," *J. Exper. Theoret. Artif. Intell.*, pp. 1–19, Jan. 2023, doi: https://doi.org/10.1080/0952813X.2023.2165714

[26] F. S. Alotaibi and V. Gupta, "A cognitive inspired unsupervised language-independent text stemmer for Information retrieval," *Cogn. Sys. Res.*, vol. 52, pp. 291–300, Dec. 2018, doi: https://doi.org/10.1016/j.cogsys.2018.07.003

[27] M .E. Basiri and A. Kabiri, "HOMPer: A new hybrid system for opinion mining in the Persian language," *J. Info. Sci.*, vol. 46, no. 1, pp. 101–117, 2020, doi: https://doi.org/10.1177/0165551519827886

[28] P. Vaishali Kadam, B. Kalpana Khandale, and C. Namrata Mahender, "Design and development of marathi word stemmer," in *Proc. Second Int. Conf. Adv. Comput. Eng. Commun. Syst.*, Springer, 2022, pp. 35–48, doi: https://doi.org/10.1007/978-981-16-7389-4_4

[29] M. V. Raju and M. Sreenivasulu, "A Lightweight Stemmer for Telugu Languag," *4th Int. Conf. Inventive Res. Comput. Appl.*, IEEE, Sep. 21–23, 2022, pp. 1385–1388, doi: https://doi.org/10.1109/ICIRCA54612.2022.9985623

[30] A. A. Sattar, S. Abbasi, M .U. Rahman, A. Baig, and M. Nizamani, "Sindhi stemmer using affix removal method," *Int. J. Adv. Trend. Comput. Sci. Eng.*, vol. 10, no. 3, pp. 2447–2451, 2021.

[31] H. Kaur and P. K. Buttar, "A rule-based stemmer for Punjabi adjectives," *Int. J. Adv. Res. Comput. Sci.*, vol. 11, no. 6, pp. 15–19, 2020, doi: http://dx.doi.org/10.26483/ijarcs.v11i6.6665

[32] S. Das, R. Pandit, and S. K. Naskar, "A rule based lightweight Bengali stemmer," *Proc. 17th Int. Conf. Nat. Lang. Process.*, 2020, pp. 400–408, doi: https://aclanthology.org/2020.icon-main.55

[33] K. T. P. M. Kariyawasam, S. Senanayake, and P. S. Haddela, "A rule based stemmer for Sinhala language," in *14th Conf. Indust. Info. Sys.*, IEEE, Dec. 18–20, 2019, pp. 326–331, doi: https://doi.org/10.1109/ICIIS47346.2019.9063286

[34] L. Sarmah, S. K. Sarma, and A. K. Barman, "Development of Assamese rule based stemmer using WordNet," in *Proc. 10th Global WordNet Conf.*, Wrolac, Poland, 2019, pp. 135—-139.

[35] A. A. Suryani, D. H. Widyantoro, A. Purwarianti, and Y. Sudaryat, "The rule-based sundanese stemmer," *ACM Trans. Asian Low-Resource Lang. Info. Process.*, vol. 17, no. 4, pp. 1–28, 2018, doi: https://doi.org/10.1145/3195634

[36] M. Ali, S. Khalid, and H. M. Aslam, "Pattern based comprehensive urdu stemmer and short text classification," *IEEE Access,* vol. 6, pp. 7374–7389, Dec. 2017, doi: https://doi.org/10.1109/ACCESS.2017.2787798

[37] J. Sheth and B. Patel, "Dhiya: A stemmer for morphological level analysis of Gujarati language," in *Int. Conf. Issues and Challeng. Intell. Comput. Techniq.*, IEEE, Ghaziabad, India, Feb. 7–8, 2014, pp. 151–154, doi: https://doi.org/10.1109/ICICICT.2014.6781269

[38] A. Paul, A. Dey, and B. S. Purkayastha, "An affix

removal stemmer for natural language text in nepali," *Int. J. Comput. Appl.,* vol. 91, no. 4, pp. 1–4, 2014.

[39] P. Koirala and A. Shakya, "A Nepali Rule Based Stemmer and its performance on different NLP applications," *arXiv preprint arXiv:2002.09901,* 2020, doi: https://doi.org/10.48550/arXiv.2002.09901

[40] R. A. Baeza-Yates, "Text-Retrieval: Theory and Practice.," in *Proc. IFIP 12th World Comput. Cong. Algorith. Software Architec. Inform. Process '92*, 1992, pp. 465–476.

[41] M. Melucci and N. Orio, "A novel method for stemmer generation based on hidden Markov models," in *Proc. 12th Int. Conf. Inform. knowledge Manag.*, 2003, pp. 131–138, doi: https://doi.org/10.1145/956863.956889

[42] P. Majumder, M. Mitra, S. K. Parui, G. Kole, P. Mitra, and K. Datta, "YASS: Yet another suffix stripper," *ACM Trans. Info. Sys.*, vol. 25, no. 4, pp. 18–es, Oct. 2007, doi: https://doi.org/10.1145/1281485.1281489

[43] T. Anzai and A. Ito, "Recognition of utterances with grammatical mistakes based on optimization of language model towards interactive CALL systems," in *Proc. of 2012 Asia Pac. Signal Info. Process. Associ. Ann. Summit Conf.*, California, USA, Dec. 3–6, 2012, pp. 1–4.

[44] A. Ali, A. Hussain, and M. K. Malik, "Model for english-urdu statistical machine translation," *World Appl. Sci.,* vol. 240. no. 10, pp. 1362–1367, 2013, doi: https://doi.org/10.5829/idosi.wasj.2013.24.10.760

[45] A. Jabbar, S. ul Islam, S. Hussain, A. Akhunzada, and M. Ilahi, "A comparative review of Urdu stemmers: Approaches and challenges," *Comput. Sci. Rev.*, vol. 34, Art. no. 100195, Nov. 2019, doi: https://doi.org/10.1016/j.cosrev.2019.100195

[46] C. D. Patel and J. M. Patel, "Influence of GUJarati STEmmeR in Supervised Learning of Web Page

Categorization," *Int. J. Intell. Sys. Appl.*, vol. 13, no. 3, pp. 23–34, 2021, doi: https://doi.org/10.5815/ijisa.2021.03.03

[47] V. Giri, "MTStemmer: A multilevel stemmer for effective word pre-processing in Marathi," *Turk. J. Comput. Math. Educ.*, vol. 12, no. 2, pp. 1885–1894, 2021, doi: https://doi.org/10.17762/turcomat.v12i2.1527

[48] H. Alshalabi, S. Tiun, N. Omar, F. N. AL-Aswadi, and K. A. Alezabi, "Arabic light-based stemmer using new rules," *J. King Saud Univ.-Comput. Info. Sci.*, vol. 34, no. 9, pp. 6635–6642, Oct. 2022, doi: https://doi.org/10.1016/j.jksuci.2021.08.017

[49] R. Kansal, V. Goyal, and G. S. Lehal, "Rule based urdu stemmer," in *Proc. COLING 2012: Demonstration Papers*, 2012, pp. 267–276.

[50] S. D. Patel, J. M. Patel, "GUJSTER: a Rule based stemmer using Dictionary Approach," in *Int. Conf. Inv. Commun. Comput. Technol.*, Coimbatore, India, Mar. 10–11, 2017, pp. 496–499, doi: https://doi.org/10.1109/ICICCT.2017.7975249

[51] A. Mateen, M. K. Malik, Z. Nawaz, H. M. Danish, M. H. Siddiqui, and Q. Abbas, "A hybrid stemmer of punjabi shahmukhi script," *Int J Comput Sci Netw Secur,* vol. 17, no. 8, pp. 90–97, Aug. 2017.

[52] A. Rahimi, "A new hybrid stemming algorithm for Persian," *arXiv preprint arXiv:1507.03077,* 2015, doi: https://doi.org/10.48550/arXiv.1507.03077

[53] M. Hadni, S. A. Ouatik, and A. Lachkar, "Effective Arabic stemmer based hybrid approach for Arabic text categorization," *Int. J. Data Mining Knowledge Manag. Proc. Acad. Indus. Res. Collabo. Center*, vol. 3, no. 4, pp. 1–4, July 2013, doi: https://doi.org/10.5121/ijdkp.2013.3401

[54] K. Suba, D. Jiandani, and P. Bhattacharyya, "Hybrid inflectional stemmer and rule-based derivational stemmer for gujarati," in *Proc. 2nd Workshop South Southeast Asian Nat.*

Language Process., 2011, pp. 1–8.

[55] H. Sak, A. W. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," *Interspeech*, pp. 338–342, 2014.

[56] M. S. Islam, S. S. S. Mousumi, S. Abujar, and S. A. Hossain, "Sequence-to-sequence Bangla sentence generation with LSTM recurrent neural networks," *Proc. Comput. Sci.*, vol. 152, pp. 51–58, 2019, doi: https://doi.org/10.1016/j.procs.2019.05.026

[57] F. Chollet, " A ten-minute introduction to sequence-to-sequence learning in Keras." The Keras Blog. https://blog.keras.io/a-ten-minute-introduction-to-sequence-to-sequence-learning-in-keras.html (Accessed Dec. 3, 2022).