AN ALGORITHM FOR REMOVING MORPHOLOGICAL VARIATIONS OF WORDS FOR NLP APPLICATIONS

Rahim Bux[1], Mohsin Ali Memon[2], Amirita Dewani[3]

Mehran University of Engineering & Technology, Jamshoro, PAKISTAN.

[1]raheembux991@gmail.com, [2] mohsin.memon@faculty.muet.edu.pk, [3]amirita@faculty.muet.edu.pk

ABSTRACT

In recent past years, the field of natural language processing has been steadily growing due to the popularity of the Internet. It is believed that the morphological forms of the words have the same basic meaning and should be assigned to the same stem. In Natural Language processing functions normalization is a very common requirement. It is very important in most Information Retrieval systems. Normalization involves stemming to restore the various grammar/word forms of a word to their root form. This work is addressing the problem of dealing with morphological variants of words that have similar semantic interpretations written in Urdu language (Latin script) which is a resource-poor language. This works aims to reduce inflectional forms (Sometimes called derivationally related forms) of a word to a common base form. The main goal of this work will be to propose, implement, and evaluate a Truncating algorithm for removing the suffixes or prefixes (commonly known as affixes) of a word. The algorithm is based on the formation of linguistic rules for the truncation of words.

Keywords: NLP, Normalization, Urdu Script, morphemes, Affixes

INTRODUCTION

In natural language processing, the core step to process and classify any textual data is stemming [4]. Stemming is the process to extract the root from the given word, it is also used for the sentiment analysis of text [2]. it is the process of getting the root word from the given word. For achieving that phenomena stemmers are used to get the root word from the different forms of the word [9]. These stemmers use different techniques like prefix removing, suffix removing and some predefined rules and dictionaries [12]. For example, we have the words information, informative, and information these words are derived from the same word so if we are wanted to extract the root word for these words than by removing the last characters i.e. postfix the root word can be extracted. So, the base word will be “inform”.

Although stemming has several challenges to implement in different languages [8]. But to achieve this task different algorithms are developed in different natural languages. A good number of stemmers are present in English because it is frequently used and does not have much complex grammatical structure like other languages Urdu, Arabic, Hindi, etc. while research is being done on the other languages also. Lovins was the first who develop the stemmer for the English language in 1968 [9]. It was a rule-based stemmer that using the conditions to remove the suffix of the word. Several other stemmers are developed for the English language, porter stemmer is one of the most well-known and often used. These stemmers are very commonly used in information retrieval applications and text mining [6].

ISSN: 2223-9944, e ISSN: 2223-9553

In our research, we will develop the stemmer for the Urdu language (Latin script) because there is no significant work is done for the stemmer of Urdu language (Latin script) and there is a lot of public as well private data available in it which can be very useful in many applications. we will use the hybrid approach for developing the algorithm. The approach includes the two methods for achieving the task, one is rule-based in which we will define the rules for the processing of any word to get the root word from it, and another method is dictionary-based, in which we will use the dictionary of different words for the specific words like names.

RELATED WORK

Stemmers are developed by using different techniques, some of the stemmers are rule-based, some are dictionary and others can be a hybrid approach based it depends on the complexity and morphological form of data [5]. A considerable amount of work has been done on the design and development of stemmers in a different language. 1980s Porter extends the work of the Lovins stemming algorithm for the English language. The effective and efficient rule-based algorithm was developed by Porter in which he reduced to Lovins 264 rules to just 60 rules and he got good results. this algorithm is frequently in different natural language processing libraries till today.it uses the five steps to remove the suffix of any word [9]. Paice/Husk develop the iterative algorithm for stemming it comprises about 120 rules that are indexed by the suffix of the last letter of the word. The main advantage of that algorithm is that it removes the unnecessary characters and replace the necessary one where needed [9]. Dawson also gives the extension of the Lovins algorithm it covers a large number of list of suffixes that is about 1200. The core advantage of the algorithm is to use more suffixes [9].

Research has been also done the languages other than English, like Belal Abuata and Asma Al-Omari develop a rule-based stemmer for the Arabic gulf dialect [1]. The approach used for the algorithm is rule-based they created the set of rules defined in the form of an algorithm to remove the suffixes and prefixed dialect words, the testing results show the overall accuracy of 88% for this stemmer. Superior root-based stemmer was also developed by the khoja and Garside for the Arabic language [11]. The algorithm divides the word into prefix, suffix, and infix, then matches with the affix and get the result. Vaishali Gupta and her co-authors [3] developed the rule-based stemmer for the Urdu language in which they create the list of 119 affix rules which are used to compare to word to get expected stemmed word, the evaluation results concluded the accuracy of 86.5%. A good stemmer for the Urdu language is developed by Zahid Hussain [10] they developed a dictionary-based algorithm. it uses the data from the different resources to achieve high accuracy. They achieved 94.85% of the overall accuracy for testing data. The multi-step hybrid approach is also used to develop the stemmer for the Urdu language in which research deals with different types of unigrams, bigrams, trigram, and their features. They used word corpus and text corpus to improve the performance of an algorithm and achieved an accuracy of 92.97% [7]. A comprehensive study [13] was proposed by Mubashir Ali for the morphological rich Urdu language. stemmer was developed using infix removing rules for pure Urdu words and for borrowed words (The words which are taken from other languages to Urdu). Experimental results show an average accuracy of 85.60% using different corpora. The first Rule-based stemmer for the Persian language was developed by M Tashakori [14] named Bon.

The stemmer does not use any dictionary of the Persian text, according to experimental evaluation using this stemmer up to 40% performance can be improved for any query in an information retrieval system. The affix removal approach for Gujrati text was proposed by [15]. They developed a hybrid stemmer for Gujrati language which uses affix removal rules and dictionary lookup for the development of the algorithm. with the help of specially

ISSN: 2223-9944, e ISSN: 2223-9553

devised suffix-striping substitution of words and prefix rules, they claim an accuracy of 96.63%. A big research gap has been observed in Urdu language (Latin script), very limited work is done on the Urdu language (Latin script) stemming because it has a very complex structure due to no grammatical form and it is not a formal language. so, we have chosen this topic as research work for the development of an effective stemmer for the Urdu language (Latin script).

METHODOLOGY

The development of the algorithm was divided into different phases, where we had collected the data from the social networks, then analyze the data to build the dataset. After collecting our dataset, we had created the basic naming words dictionary and rules for the development of the algorithm. The research phases are defined below:

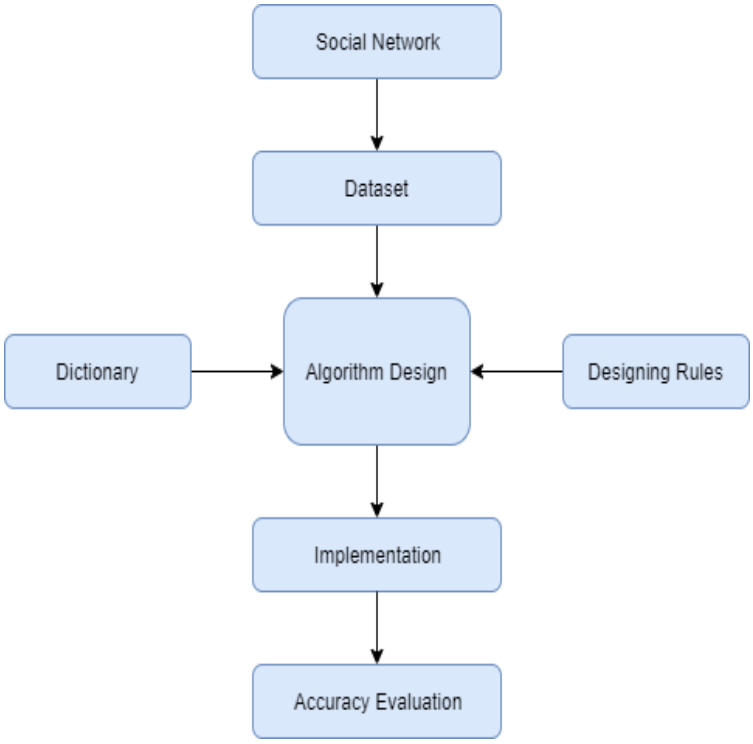

Figure 1. Methodology Diagram

DATASET COLLECTION

In this phase, we had collected the data from different sources like the social network site Twitter from where we had extracted the data from different pages using Twitter API and analyze the data to build the dataset for creating the rules for the algorithm. Besides that, we had also collected the data from Kaggle which is used for the testing purpose of the algorithm.

DICTIONARY CREATION

Our algorithm truncates the words by removing its postfix or prefix to extract the word. But some naming words should be excluded from the truncation process because the name itself can be considered as the root word. So, for that, we had created the list for the naming words which contains the most frequently used names which should not be truncated more during the truncation process.

ISSN: 2223-9944, e ISSN: 2223-9553

POSTFIX LIST

After collecting our data, we analyze it so that how we can find out the postfix list. We collected almost a list of 10000 different words which are extracted from different pages and used them to find out the postfix of the words and we created the list of postfixes which will be used for making the rules for the truncation of words. Table1 shows some example postfix list along with the root word.

Table 1. Sample postfix list

Word	Postfix	Stem word
zaroratmand	mand	zarorat
Khabarnaak	naak	khabar
chupana	ana	chup
pardadaar	daar	parda

DEFINING POSTFIX RULES

For removing the postfix of the word's rules are made according to the postfix and morphological structure of the words. These postfix rules are used for removing the postfix of the word to filter out the root word. This process requires some analysis on the words that how postfix can be extracted from the words because Urdu text has a complex morphological structure which leads to making the word meaningless after removing its postfixes.

PREFIX LIST

In this phase, we had separated the starting characters of words i.e. prefixes from the words and created a prefix-list which is used for making the rules for the prefix removal of the words. List contains prefix like bad which is a prefix of the words like (badikhlaq, badguman, etc). Table 2 below contains come to list sample prefix list.

Table 2. Sample prefix list

Word	Prefix	Stem word
baaakhtiar	Baa	akhtiar
badikhlaaq	Bad	ikhlaaq
khushnaseeb	Khush	naseeb
Naaihill	Naa	ihill

DEFINING PREFIX RULES

This phase focuses on creating rules according to the prefix list. These are applied to the words to extract the root word. So, when we input a word into the algorithm it will check the word if it has a prefix which is defined in prefix-list then it removes that prefix and returns the word.

IMPLEMENTATION

We had implemented our algorithm using python and its libraries for text preprocessing. This process involves the rules implementation and naming dictionary lookup for the word. Pseudo Code for the algorithm to extract root word is below:

ISSN: 2223-9944, e ISSN: 2223-9553

Step 1: Input Word
Step 2: Check Word Length
    If Word Length>4
        Go to Step 3
    Else:
    Returned Word is basic Word.
Step 3: Check Word in Naming and Stem Word Lists
    If Word Not in Stem Words List and Word Not in Naming Dictionary
        Go to Step 4
    Else
        Returned Word is basic Word.
Step 4: Check Postfix of Word
    If Word Has Postfix from Postfix List
        Apply Corresponding Rule
        Return Word
    Else
        Go to Step 5.
Step 5: Check Prefix of Word
    If Word Has Prefix from Prefix List
        Apply Corresponding Rule
        Return Word
    Else
        Go to Step 6.
Step 6: Apply Cosine Similarity
    Return Word.

TESTING AND RESULTS

We had tested our algorithm against the dataset which is publicly available at kaggle. our testing data contains the 18068 unique words which were tested against our algorithm. The overall testing results are shown in table3. Total 11924 words are correctly stemmed by our algorithm. Though there were some stems that were not present in the list. The analysis shows that major incorrect stems were due to the prefix because its morphology is more complex than prefix there were many cases where some word is changed completely and lost its meaning due to the prefix removal. The contribution rate of correct stemmed by each technique is also mentioned where 6120 words are correctly converted in morphemes using prefix and postfix rules. 3303 words are correctly converted in morphemes by cosine similarity. 2501 words are correctly stemmed due to the dictionary approach these words are nouns or already present in the stemmed list and they do not need to stem further.

Table 3. Overall results

Testing Results	Values
Total Number of words tested	18068
Correctly converted	11924
Incorrect converted	6144
Correct converted by Rules	6120
Correct converted by Dictionary	2501
Correct converted by Cosine Similarity	3303

ISSN: 2223-9944, e ISSN: 2223-9553

ACCURACY CHART

The overall testing of the algorithm shows the accuracy of 65.99%. Figure2 shows the performance of different techniques used by our algorithm to get correct root words. 51.3% of the correct stemmed are given by the prefix and postfix rules. While the cosine similarity extracts the 271.7% correct root words against the given data. The remaining 21% are the words which do not need to further stem because they may be naming word or already stemmed so our algorithm did not apply rules and cosine similarity on such type of words.

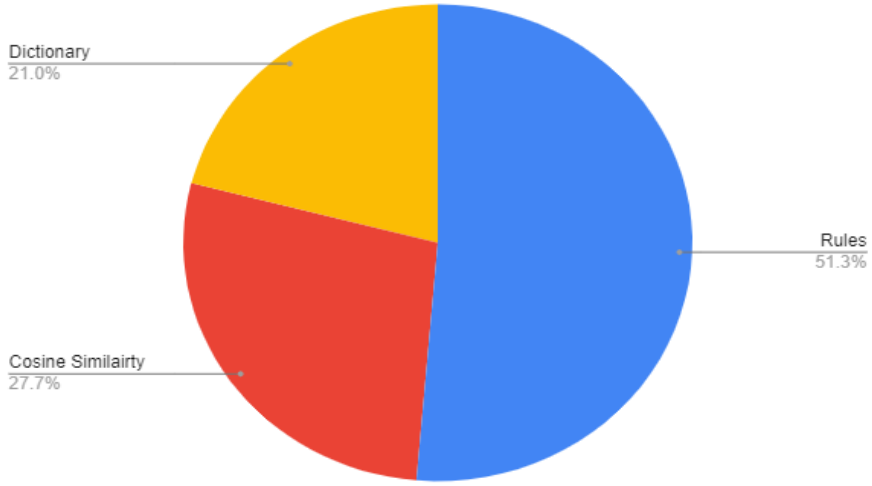

Figure 2 Correct Stem Chart

ACCURACY PRECISION RECALL CHART

Figure 3 shows the overall accuracy, precision, and recall of the algorithm. Where accuracy is calculated using equation1, while recall and precision are calculated by equation 2 and 3, respectively. The results show that the accuracy of the algorithm is 65.99% while the recall is 79.3%. As our algorithm is developed in such a way that it always gives the stem of the word and there will be not any case were not going into the process of extracting the root word then there will be no false positive so the precision of algorithm will 100%.

$$Accuracy(\%) = \frac{Tp + Tn}{Tp + Tn + Fp + Fn} \times 100 \qquad (1)$$

$$Recall(\%) = \frac{Tp}{Tp + Fn} \times 100 \qquad (2)$$

$$Precision(\%) = \frac{Tp}{Tp + Fp} \times 100 \qquad (3)$$

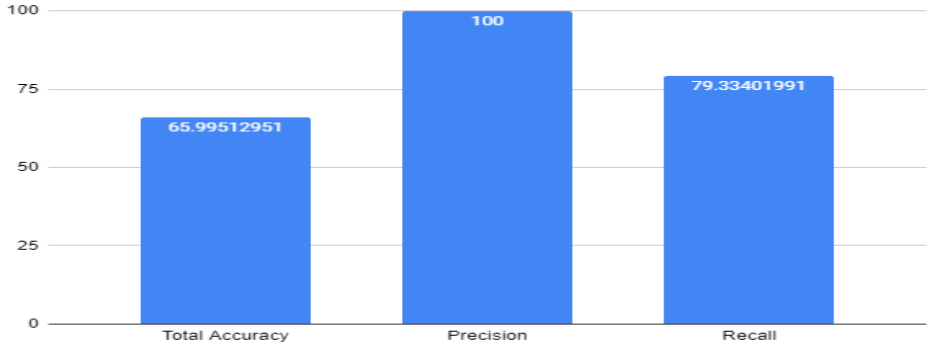

Figure 3. Accuracy precision recall chart

RESULT CHART

The performance of the algorithm is measured against the given dataset. Figure 4 shows the performance of the algorithm in the percentage of extracting the correct stem of the words and the incorrect stem produced by the algorithm. The performance of giving the correct stemmed is 66%. while 34 % of the words are incorrectly stemmed by the algorithm.

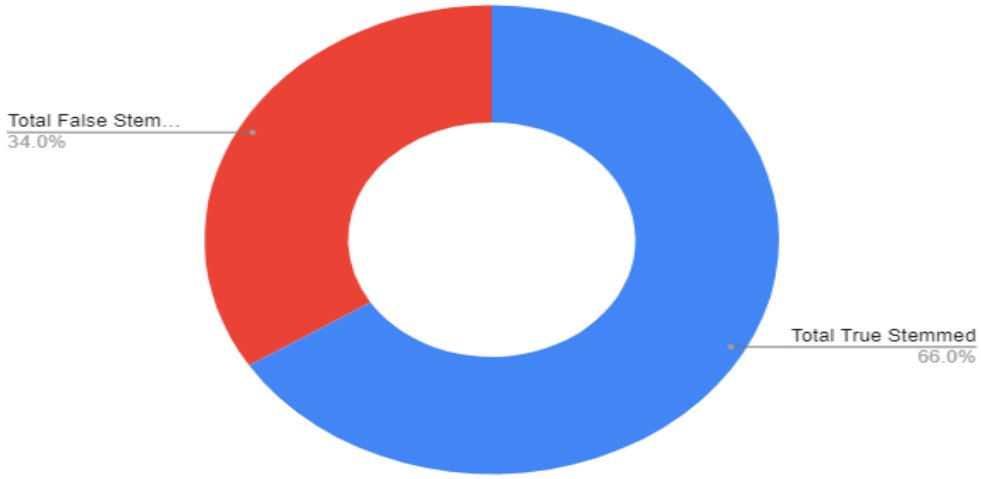

Figure 4. Result Chart

RECALL AND PRECISION PLOT

Precision and recall on different intervals of testing data are plotted in figure 5 where the results show the increase in the testing data algorithm is performing better results.

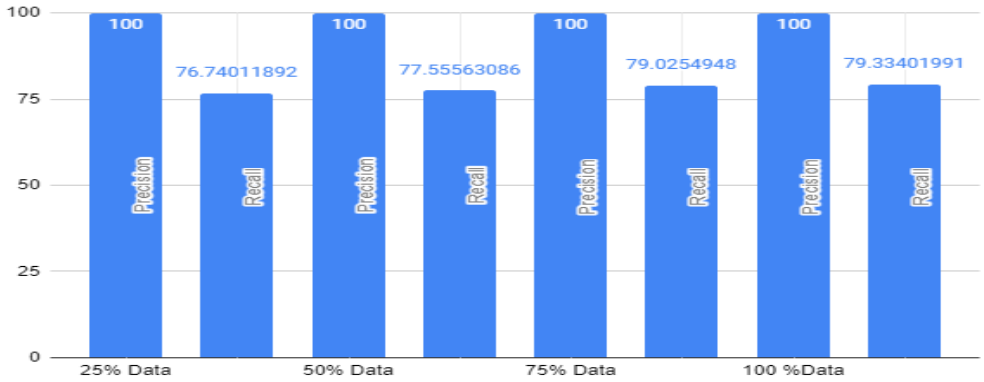

Figure 5. Recall precision intervals chart

CONCLUSION

From the overall evaluation of the algorithm, we can conclude that though there are many challenges to deal with the Urdu script due to complex morphological variations in words and different writing styles. but still, the algorithm is showing satisfactory results and we can improve the results by modifying and increasing the rules and by improving the evaluation data.

REFERENCES

Abuata B., & Al-Omari, A. (2015). A rule-based stemmer for Arabic Gulf dialect. Journal of King Saud University - *Computer and Information Sciences, 27*(2), 104–112. DOI: 10.1016/j.jksuci.2014.04.003.

Ali, M., Khalid, S., & Aslam, M. H. (2018). Pattern Based Comprehensive Urdu Stemmer and Short Text Classification. *IEEE Access*, *6*, 7374–7389. DOI: 10.1109/access.2017.2787798

Bijal Dalwadi, Nikita Desai (2016). *An affix removal stemmer for Gujrati text.* International Conference on Computing for Sustainable Global Development 978-9-3805-4421-2/16/$31.00 c 2016 IEEE.

Flores, F. N., & Moreira, V. P. (2016). Assessing the impact of Stemming Accuracy on Information Retrieval – A multilingual perspective. *Information Processing & Management, 52*(5), 840–854. DOI: 10.1016/j.ipm.2016.03.004

Gupta, V., Joshi, N., & Mathur, I. (2013). *Rule based stemmer in Urdu.* 2013 4th International Conference on Computer and Communication Technology (ICCCT). DOI: 10.1109/iccct.2013.6749615

Jabbar, A., Iqbal, S., Akhunzada, A., & Abbas, Q. (2018). An improved Urdu stemming algorithm for text mining based on multi-step hybrid approach. *Journal of Experimental & Theoretical Artificial Intelligence, 1*–21. doi: 10.1080/0952813x.2018.1467495

Kassim, M. N., Maarof, M. A., Zainal, A., & Wahab, A. A. (2016). *Word stemming challenges in Malay texts: A literature review.* 2016 4th International Conference on Information and Communication Technology (ICoICT). DOI: 10.1109/icoict.2016.7571887

Anjali Ganesh Jivani (2011). A Comparative Study of Stemming Algorithms. 2011 *Int. J. Comp. Tech. Appl., 2* (6), 1930-1938.

Mubashir Ali, Shehzad Khalid, and Muhammad Saleem (2019). Comprehensive Stemmer for Morphologically Rich Urdu Language. *The International Arab Journal of Information Technology, 16*(1), January 2019.

Noor, F., Bakhtyar, M., & Baber, J. (2019). *Sentiment Analysis in E-commerce Using SVM on Roman Urdu Text. Lecture Notes of the Institute for Computer Sciences,* Social Informatics and Telecommunications Engineering Emerging Technologies in Computing, 213–222. DOI: 10.1007/978-3-030-23943-5_16

S. Khoja and R. Garside, (1999). *Stemming Arabic Text",* Lancaster, UK, Computing Department, Lancaster University.

Sun, S., Luo, C., & Chen, J. (2017). A review of natural language processing techniques for opinion mining systems. *Information Fusion, 36*, 10–25. DOI: 10.1016/j.inffus.2016.10.004.

Swain, K., & Nayak, A. K. (2018). *A Review on Rule-Based and Hybrid Stemming Techniques.* 2018 2nd International Conference on Data Science and Business Analytics (ICDSBA). DOI: 10.1109/icdsba.2018.00012

Tashakori, M., Meybodi, M., & Oroumchian, F. (2002). *Bon: The Persian Stemmer.* Lecture Notes in Computer Science EurAsia-ICT 2002: Information and Communication Technology, 487–494. DOI: 10.1007/3-540-36087-5_57.

Zahid Hussain, Sajid Iqbal, Tanzila Saba, Abdulaziz, Almazyad , Amjad Rehman (2017). Design and Development of Dictionary-Based Stemmer for the Urdu Language. *Journal of Theoretical and Applied Information Technology 15th August 2017. 95*(15).

ISSN: 2223-9944, e ISSN: 2223-9553