ISSN 2278-3091

Volume 10, No.3, May - June 2021

International Journal of Advanced Trends in Computer Science and Engineering

Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse1331032021.pdf

https://doi.org/10.30534/ijatcse/2021/1341032021

Sindhi Stemmer using Affix Removal Method

Ambreen A. Sattar[1], Suhni Abbasi[2], Mutee U Rahman[3], Amber Baig[4] and Masroor Nizamani[5]

[1]Information Technology Centre, Sindh Agriculture University, Tandojam, Hyderabad, Pakistan, ambreen.nizamani7@gmail.com

[2]Information Technology Centre, Sindh Agriculture University, Tandojam, Hyderabad, Pakistan, suhni.abbasi@sau.edu.pk

[3]Department of Computer Science, Isra University, Hyderabad, Pakistan, muteeurahman@gmail.com

[4]Department of Computer Science, Isra University, Hyderabad, Pakistan, amberbaig@gmail.com

[5] Department of Computer Science, Isra University, Hyderabad, Pakistan, masroor342@gmail.com

ABSTRACT

Stemming is the process of mapping various inflections of a word to its base form. Stemmer is an essential component of Information Retrieval (IR) systems and different Natural Language Processing pipelines. This research reports the development and evaluation of stemmer for a resource poor language Sindhi. The stemmer is using a lexicon-based affix removal technique for stemming. The developed lexicon represents the base forms and the algorithm uses this lexicon during the affix removal process. The overall performance accuracy is evaluated, and stemmed error rate is calculated. The results show 89.57% overall performance accuracy.

Key words: Natural language processing, Information Retrieval, Sindhi, Lexicon, Yet another Suffix Stripper, Hidden Markov Model.

1. INTRODUCTION

Modern information retrieval (IR) systems make use of Natural Language Processing (NLP) techniques to improve their performance. Natural language processing (NLP) is the branch of computer science concerned with how computers can be used to understand and manipulate natural language text or speech to carry out effective operations [5]. IR deals with the processing of documents which include the text so that they can be retrieved rapidly based on the query of the user [13]. One of the NLP techniques used by IR systems is stemming. Stemmer is an essential component of almost all IR systems [1] which is used to reduce inflected words to their base forms [3-4]. Many stemmers are available for languages such as English (e.g. [10], [17]), Arabic (e.g. [2], [9]), and Urdu [8]. This research study focuses on the development of a stemmer for Sindhi language. Sindhi is a resources poor language in terms of NLP applications. Few Sindhi stemming research studies are also available and discussed in following section. However, this research study tries to improve the performance efficiency of Sindhi stemming.

Various types of stemming algorithms are there, they follow different techniques to find the base form of the inflected words. These algorithms are divided into three different categories which include: affix removal, statistical and hybrid algorithms. In affix removal stemming algorithms prefixes and suffixes are removed by applying some set of rules. While in the statistical methods, statistical information is used from a corpus or dictionary to learn morphology. N-Gram, YASS [11] and HMM (Hidden Markov Model) stemmers are examples of statistical stemmers. Prefixes and suffixes are removed after applying some statistical procedure [6]. On the other hand, hybrid methods are combination of affix removal techniques and statistical methods. The proposed Sindhi stemmer uses the affix-removal method to find out the base forms of inflected words.

Following sections discuss related work, methods and materials including stemming algorithms and different stemming scenarios, results, and conclusion.

2. RELATED WORK

Most of the classical stemming research work include different studies of developing English stemmers. For instance, Lovins Stemming Algorithm [10], which uses technical vocabularies. It identifies and removes the longest endings by using transformation rules. Another popular stemmer is Porter Stemmer [17]. The Porter stemming algorithm makes use of list of suffixes, transformation rules and context-sensitive rules. Suffix removal and word conflation algorithm [21] is another classical work which extends the Lovins Stemming Algorithm by enhancing the coverage of suffixes list. Later [22] proposed another stemming algorithm developed at Lancaster University. It uses an iterative algorithm with stored rules. Krovetz stemming algorithm [23] is another work on English stemming which uses inflectional rules of plural to singular, past tense to present tense, gerund and continuation “ING” removal, along-with some transformation rules.

Apart from classical stemming work for English discussed above various stemmers for closely related languages to Sindhi including Urdu and Punjabi are also available. For Urdu a stemming algorithm named "assas band" was proposed by [19] which uses prefix and suffix removal technique with exception lists to ensure prefix or suffix removal. Another stemmer for Urdu is Light weight stemmer for Urdu proposed by [18]. This algorithm uses rule-based approach where inflectional morphology rules are used. Prefixes and suffixes are removed however, exceptions are handled by using an exceptional list like assas band. Another research work for Urdu stemming include [8]. This stemmer was developed by using a rule-based technique. Stemmer has two main components: rules and the lexicon. To remove prefixes and suffixes from the word, rules were developed. Also, the database (lexicon) was designed which contains root words along with their frequencies. When input was given to the system, tokens were generated of the words by using delimiters as space. The text was normalized by removing special characters like.? , ' "@. After normalization prefix and suffix rules were applied to the word. If the appropriate rule, not found, then the system will return this word as a root word. However, if rule found for that particular word, then the word is further divided into possible morphemes. Afterwards, these morphemes are matched against the one which is stored in the database to find frequencies. Maximum frequency in possibilities list is looked up and word with maximum frequency is returned as root form. The accuracy of this Rule-based Urdu stemmer is 85.14%.

For Punjabi an enhanced stemmer based on suffix removal and table lookup techniques is proposed by [7], this study is applied on Gurmukhi script examples.

Among related work for Sindhi stemming [16] is one of the initial stemming studies for Sindhi. In this work first Rule-based stemming algorithm for Sindhi language is proposed. The proposed algorithm is divided into two parts, first is the lexicon and the second is developed Rules. A lexicon has a total of 50,327 words. The original word, its prefix, and suffix can be added, updated, and removed from the lexicon directly from the user interface of the developed application. Total 50,327 words stored inside the lexicon. The second part of an algorithm is Rules, there is a total of 38 Linguistic Rules developed and included in repositories. Sixteen rules are developed for prefixes and fifteen rules for suffixes. The performance accuracy of an algorithm was 84.85%.

[20] presented their stemming work for Sindhi where they proposed an algorithm for stemming. Proposed algorithm finds the root of the word by breaking it into different affixes, this root is considered the base work or stem. Affixes and suffixes are removed by using table lookup method. Accuracy results are not reported, the algorithm and some statistics of the corpus / words are presented.

Later [15] proposed an unsupervised stemming algorithm for Sindhi. The overall experimental setup includes ten thousand (10000) sentences, the algorithm, and its implementation. List of suffixes is used for suffix striping. The overall accuracy of 87% is achieved which is better than the earlier rule based stemming accuracy which was 85%. However, this stemming algorithm is tested and evaluated on Devanagari script of Sindhi.

3. METHODS AND MATERIALS

This research study comprises of the development of the stemmer by truncating the prefixes and suffixes to get the original base word form or stem. A lexicon containing root words is developed using the MySQL database, A lexicon-based algorithm is also proposed and discussed. Figure 1 shows the overall flow of proposed algorithm. Following sections discuss the overall methodology with proposed algorithm and different stemming scenarios.

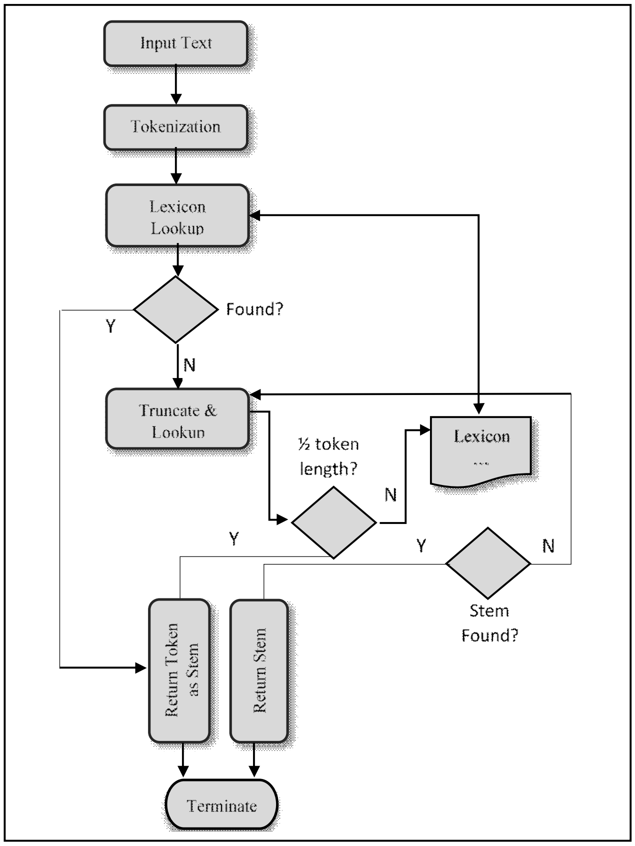


Figure 1: Overall Flow of Proposed Algorithm.

3.1 Proposed Algorithm

A lexicon-based affix removal algorithm for stemming is proposed. Base forms or stems are stored in a lexicon database. For every input word system looks that word into the stem lexicon, successful search implies that it is already in base word form, therefore the system simply returns same word as a stem. In case of unsuccessful search, it is not a stem, the algorithm will truncate the affix and keep searching the lookup table until the stem is found, or half of the word length is reached. There are different scenarios in this process. Following sections discuss different stemming scenarios with examples.

3.2 Stemming Scenarios

Different stemming scenarios can be there, for example consider a case of suffix removal where after tokenization the candidate token for stemming is “نشانيون”. Initially the stemming process will remove the last letter “ن” and resulting word “نشانيو” will be looked into the stem / root words list. In this case “نشانيو” is not a valid stem and will not be found in the lookup list. As the total length of candidate word “نشانيون” is seven, therefore, the process of removing the last letter is repeated at least three times in case of unsuccessful lookup and the successful retrieval of stem process will terminate immediately. In this case process will be repeated as “نشانيو” is not a valid stem. The last letter “و” of remaining word “نشانيو” will also be removed and resulting token “نشاني” will be looked up in the stem words list. This time “نشاني” is a valid stem therefore, search will be successful and “نشاني” will be returned as stem of “نشانيون”.

Another case may be when truncated word is not found in the list at all. For example, the token “شعبا”, when truncated will have word form “شعب”, which is invalid and will not be found in the stem list. In this case next iteration will cause word to be truncated to “شع” which is also invalid. The process of removing the last letter will now stop as half of the word is truncated. The algorithm will now start looking the minimum edit distance stem for the candidate token, which in this case will be “شعبو”. Therefore, the resulting stem “شعبو” will be returned.

While considering a prefix removal scenario same procedure is repeated as discussed above. However, instead of last letter removal a list of prefixes is searched and matching smallest prefix is removed and resulting word is searched in the stem list. For example, consider the case of word “ناپسند”. Here, “نا” will be removed and resulting word “پسند” will be returned as stem.

Words can also have prefix and suffix combinations where both prefix and suffix are attached with base word forms simultaneously. For instance, “اڻ چاٽائي”, “بي واقفيت” , and “ناپسندي” are some examples where both prefix and suffix are present. While dealing with these cases first prefixes are removed as discussed above. The remaining word forms are then applied suffix removal process on the same lines as discussed earlier.

A GUI based tool is designed to test and evaluate the proposed algorithm. Figures: 2, 3 and 4 show the sample input and output screenshots of the tool.

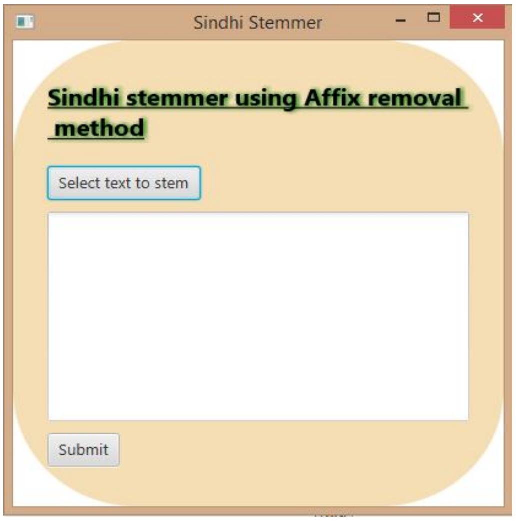
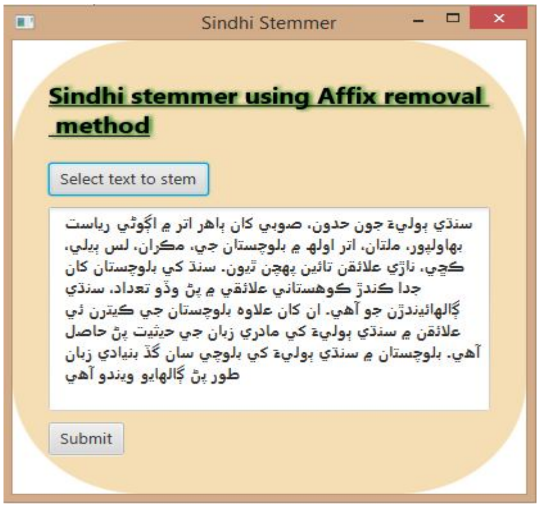
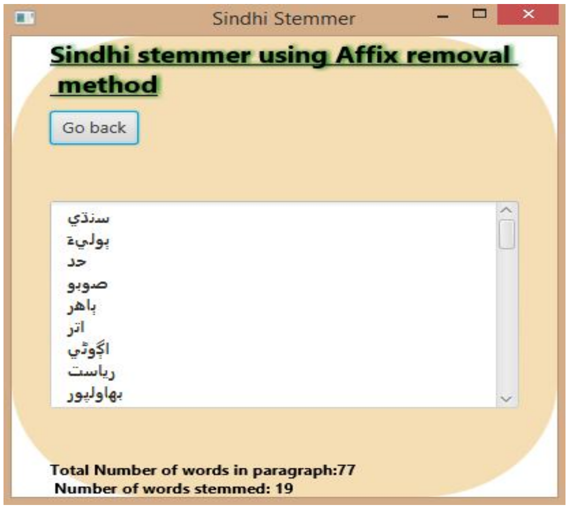


Figure 2: Input Module of Stemmer



سنڌي ٻوليءَ جون حدون، صوبي کان ٻاهر اتر ۾ اڳوڻي رياست
بهاولپور، ملتان، اتر اولهه ۾ بلوچستان جي، مڪران، لس ٻيلي،
ڪڇي، ناڙي علائقن تائين پهچن ٿيون. سنڌ کي بلوچستان کان
جدا ڪندڙ ڪوهستاني علائقي ۾ پڻ وڏو تعداد، سنڌي
ڳالهائيندڙن جو آهي. ان کان علاوه بلوچستان جي ڪيترن ئي
علائقن ۾ سنڌي ٻوليءَ کي مادري زبان جي حيثيت پڻ حاصل
آهي. بلوچستان ۾ سنڌي ٻوليءَ کي بلوچي سان گڏ بنيادي زبان
طور پڻ ڳالهايو ويندو آهي

Figure 3: Text uploaded in Input Module of Stemmer



سنڌي
ٻوليءَ
حد
صوبو
ٻاهر
اتر
اڳوڻي
رياست
بهاولپور

Total Number of words in paragraph:77
Number of words stemmed: 19

Figure 4: Output Module of Stemmer

4. RESULTS

The accuracy of the designed Sindhi stemmer is calculated by using stemmed error rate (SER). To measure the performance accuracy of the developed Sindhi stemmer, the stemmed error rate was computed. The Stemmed Error Rate (SER) is defined by [16] as given below.

$$SER = \frac{Number\ of\ Incorrectly\ Stemmed\ Words}{Total\ Number\ of\ Words} \times 100$$

For assessment, the performance of the designed stemmer, the list of words of Sindhi were collected and SER is calculated for prefixes, suffixes, and prefixes/suffixes. The calculated SER for running examples are provided in Table.1

Table 1: SER of prefixes, suffixes and prefix-suffix

Corpora	Total words Tested	Words correctly stemmed	Incorrectly Stemmed	Error Rate
Prefix	72	66	6	8.33%
Suffix	150	148	2	1.33%
Prefix/Suffix	41	34	7	17.07 %

The developed Sindhi stemmer using affix removal method has been measured on small data set having 72 prefix terms, 150 suffix terms, and 41 prefix-suffix terms. The SER of all these terms is shown in Figure 5.

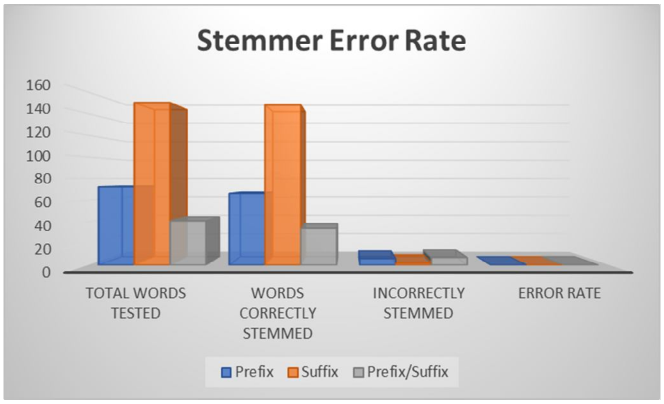


Figure 5: Stemmer Error Rate

5. CONCLUSION

Sindhi is considered a resource poor language in NLP studies. Existing research studies for Sindhi stemming are of preliminary nature where either the results are not reported or performance needs improvements. The proposed stemmer gives good results with improved performance as compared to existing stemmers. The result of the stemmer was measured using Stemmer Error Rate (SER). The performance accuracy of the developed stemmer is 91.09%. The SER of prefix words is 8.33%, SER of suffix words is 1.33% and SER of affix words is 17.07%. It is also observed that by adding more words in the lexicon, the performance accuracy of the Sindhi stemmer improved.

REFERENCES

1. A. Alnaied, M. Elbendak, and A. Bulbul, "**An intelligent use of stemmer and morphology analysis for Arabic information retrieval,**" *Egyptian Informatics Journal*, 2020.
2. A. Al-Omari, B. Abuata, and M. Al-Kabi, "**Building and benchmarking new heavy/light arabic stemmer**," in *The 4th International conference on Information and Communication systems (ICICS'13)*, 2013, pp. 17-22.
3. J. Atwan, M. Wedyan, and H. Al-Zoubi, "**Arabic text light stemmer**," *Int. J. Comput. Acad. Res*, vol. 8, pp. 17-23, 2019.
4. A. Chen and F. Gey, "**Building an Arabic stemmer for information retrieval**," in *TREC*, 2002, pp. 631-639.
5. J. Hirschberg and C. D. Manning, "Advances in natural language processing," *Science*, vol. 349, pp. 261-266, 2015.
6. A. G. Jivani, "**A comparative study of stemming algorithms**," *Int. J. Comp. Tech. Appl*, vol. 2, pp. 1930-1938, 2011.
7. G. Joshi and K. D. Garg, "**Enhanced version of Punjabi stemmer using synset**," *International Journal*, vol. 4, 2014.
8. R. Kansal, V. Goyal, and G. S. Lehal, "**Rule based urdu stemmer**," in *Proceedings of COLING 2012: Demonstration Papers*, 2012, pp. 267-276.
9. S. Khan, W. Anwar, U. Bajwa, and X. Wang, "**Template Based Affix Stemmer for a Morphologically Rich Language**," *International Arab Journal of Information Technology (IAJIT)*, vol. 12, 2015.
10. J. B. Lovins, "**Development of a stemming algorithm**," *Mech. Transl. Comput. Linguistics*, vol. 11, pp. 22-31, 1968.
11. P. Majumder, M. Mitra, S. K. Parui, G. Kole, P. Mitra, and K. Datta, "**YASS: Yet another suffix stripper**," *ACM transactions on information systems (TOIS)*, vol. 25, pp. 18-es, 2007.
12. J. Mehrad and S. Berenjian, "**Providing a Persian language singular-stemmer system (RICeST Stemmer),**" 2011.
13. C. Moral, A. de Antonio, R. Imbert, and J. Ramírez, "**A survey of stemming algorithms in information retrieval,**" *Information Research: An International Electronic Journal*, vol. 19, p. n1, 2014.
14. B. Nathani, N. Joshi, and G. Purohit, "**A Rule Based Light Weight Inflectional Stemmer for Sindhi Devanagari Using Affix Stripping Approach**," in *2018 3rd International Conference and Workshops on Recent Advances and Innovations in Engineering (ICRAIE)*, 2018, pp. 1-4.
15. B. Nathani, N. Joshi, and G. Purohit, "**Design and Development of Unsupervised Stemmer for Sindhi**

Language," *Procedia Computer Science,* vol. 167, pp. 1920-1927, 2020.

16. Shah, M., Shaikh, H., Mahar, J., and Mahar, S. "**Stemmer for information retrieval system using rule-based stripping approach**,". Sindh University Research Journal-SURJ (Science Series), vol. 48, No.4, 2016.
17. P. Willett, "**The Porter stemming algorithm: then and now**," *Program,* 2006.
18. S. A. Khan, W. Anwar, U. I. Bajwa, and X. Wang, "**A light weight stemmer for Urdu language: a scarce resourced language,**" in *Proceedings of the 3rd Workshop on South and Southeast Asian Natural Language Processing*, 2012, pp. 69-78.
19. A. Naseer and S. Hussain, "**Assas-Band, an affix-exception-list based Urdu stemmer,**" in *Proceedings of the 7th Workshop on Asian Language Resources (ALR7)*, pp. 40-47, 2009
20. M. A. Dootio and A. I. Wagan, "**Computer and Information Sciences**," 2017.
21. Dawson, J. "**Suffix removal and word conflation,**" ALLC bulletin, vol.2, No.3, pp.33-46, 1974.
22. Paice, C. D. "**Another stemmer,"** In ACM Sigir Forum Vol. 24, No. 3, pp. 56-61. New York, USA: ACM, 1990.
23. R. Krovetz. "**Viewing Morphology as an Inference Process,**" In Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, New York, pp 191-202, 1993