Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 167 (2020) 1920–1927

Procedia
Computer Science

www.elsevier.com/locate/procedia

International Conference on Computational Intelligence and Data Science (ICCIDS 2019)

Design and Development of Unsupervised Stemmer for Sindhi Language

Bharti Nathani[a],Nisheeth Joshi[b], G.N. Purohit[c]*

[a]BhartiNathani Banasthali Vidyapith, Banasthali 304022 India
[b]Nisheeth Joshi Banasthali Vidyapith Banasthali 304022 India
[c]G.N.Purohit Banasthali Vidyapith Banasthali 304022 India

Abstract

Stemmer is a fundamental NLP tool which performs the task of normalization (i.e. to remove suffixes) of inflected word. This paper presents a stemmer, design and developed for Sindhi Language, using unsupervised approach. Suffixes are extracted using "Linguistica 5 "[22] a tool for unsupervised learning of morphology. The raw corpus of 10000 sentences of Sindhi Language is used for extraction of suffixes. Unsupervised stemmer is evaluated using Direct approach. Results are compared with existing rule-based, stemmer [32] and Lemmatizer[33], 1000 words are extracted from Sindhi Dictionary for evaluation.

*Keywords:*Unsupervised Stemmer, Sindhi, Devanagari, Morphology, Rule -Based, Minimum Description Length, Linguistica 5.

* Corresponding author. Tel.: 9887078384; fax: +0-000-000-0000 .
E-mail address: bhartinathani@rediffmail.com

10.1016/j.procs.2020.03.212

1. Introduction

There are 7000 Languages in this world. Many of them are resource Poor Languages i.e. they are not equipped with Language technology tools and can be extinct soon. To prevent them, we need to sure their digital existence. In this connection we have identified Sindhi as a resource poor language of, south Asian region. Sindhi Language is included in 8th schedule of, the 22nd amendment of Indian constitution. The 8th schedule describes the languages which need to be preserved and develop. It is one of the official languages of India.

There are so many writing scripts of Sindhi Language, some of them are extinct. Two scripts Perso-Arabic and Devanagari script, are official scripts of India. In last few years lot of research is going on in Sindhi Language but all work is in Perso -Arabic Script. Sindhi Devanagari is more resource poor than Sindhi Perso-Arabic.

This paper presents the design and development of Stemmer for Sindhi Language in Devanagari script using unsupervised approach. Stemmer and Lemmatizer are two fundamental morphological analyzers. The task of Stemmer is to normalize an inflected word. The normalized word form is used in various applications of NLP such as in Information Retrieval, Text summarization, Machine translation and Topic Identification.An inflected word is normalized by removing its suffixes. Sindhiis a morphologically rich language i.e. a word can have in different form. A rule-based stemmer is available for Sindhi Language. Rule based approach requires manual extraction of suffix rule and needs language expertise. This paper presents and an unsupervised approach i.e. automatic learning of morphology to develop stemmer. This is language independent approach. With this approach it is possible to extract maximum suffixes for highly inflected language.

The remaining paper is organized in 6 sections. The Second Section define previous work in stemming. Third section describes the methodology. In Fourth section we describe our experimental work i.e. data, algorithm and evaluation. Fifth section discuss the results and Errors of system. Sixth section concluded with major findings with future research work.

2. Related Work

Stemming approaches (For Indian Languages) are classified in (1) Rule Based and (2) Machine Learning approaches. Machine Learning approach are further classified as Supervised and Unsupervised approach.[43].

In early days mostly stemmer was developed using rule-based approach. The first rule based stemming algorithm was developed by Lovins[23] for English language, which was modified by Porter [38]. The Porter stemmer remove approx. 60 suffixes in two steps.

2.1 Stemming for various Indian Languages

For Indian languages the first stemmer was developed by Ramanathan and Rao [40]. They manually extracted the suffix list and used longest matching approach for stripping of an inflected word. A Heavy weight Derivational Stemmer for Guajarati Language was proposed by Suba et. al.[45] by using rule-based approach and got the accuracy of 70.7%. They also developed an Inflectional Stemmer by using hybrid approach with 90.7% accuracy. 'Usal' an Inflectional and derivational rule-based stemmer for Urdu Language was develop by Vaishali et al. [12] and got the accuracy of 89.66%. Saharia, N. et.al. [42] Proposed a method in which the rule-based methods combine with HMM to increase the quality of Stemming. This approach was applied on Assamese Language a resource poor language of North-Eastern part and achieves the 92% accuracy.

Goldsmith [8] has discussed the algorithm for unsupervised learning of morphology. For this the Minimum Description Length Analysis algorithm was used, described in Goldsmith 2001[7]. Pandey et.al.[36] developed an approach for unsupervised stemming of Hindi Language. Evaluation was done on manually extracted test data set of Hindi WordNet data base. The training data was extracted from EMILE corpus, and consisting of 106403 words andsystem has got 89.9% accuracy. These results concluded that their stemmer outperforms the other Hindi stemmers also shows the effectiveness of this stemmer in reducing the size of index.

2.2 Related work in Unsupervised Stemming

Mudassar M. Majgaonker[27] design a rule-based stemmer and unsupervised stemmer for Marathi Language and compared the performance on a manually stemmed 1500 words test dataset. Gupta et.al.[10] have proposed a hybrid unsupervised stemming. Lemmatization approach was used to improve the performance of unsupervised stemming. Lemmatization improves the performance of system by avoiding over-stemming. Mohd. Shahid Husain [15] proposed a unsupervised stemmer for Urdu and Marathi Language. For training of system, corpus of CRULP and Marathi was used. Some suffix ruleswere generated by using frequency-based stripping and length-based stripping approach. System was evaluated on 1200 words of Emile corpus. For Urdu Language they this system has got maximum 85.36% accuracy, with frequency-based suffix generation approach. Where as in Marathi Language length-based suffix stripping gives the maximum accuracy of 82.5%.

Krishn, et.al.[20],their work is based on the learning of morphology of Hindi Language through an "automatic morphological analyser" Linguistica. Results were analyses and apply some heuristics to include some cases, in which root morpheme are changed in their morphological variants. Bhat, S. [3]. apply various techniques of unsupervised morphological segment detection in a Kannada. The algorithm was train on corpus of 990K words, and got the F-measure of 73% and concluded that algorithm gives better results on nouns in compare to verb. Can, B et.al. [5] provide a survey of various (ULM) Unsupervised Learning Morphology approaches. They describe three main category of algorithms 1. Based on Minimum description Length 2. Based on Maximum Likelihood estimation (MLE), and 3. Maximum a posteriori (MAP). This work also discusses the evaluation schemes for these algorithms.

2.3 Related work in low resource language

Saharia, N. et.al. [43], inthisresearch work focuses on stemming of resource poor Eastern Indian languages such as Bodo Manipuri and Assamese. A rule-based suffix removal approach was developed and a dictionary of frequent words was added to overcome over and under stemming errors. Most of the languages is having single letter suffixes, which create problem in stemming. To solve this problem, the HMM based hybrid approach was introduced. By using this approach, 94% accuracy is achieved for Bengali and Assamese and 82% for Bodo and 87% for Manipuri. A comparison was made of this work with Mofessor.

2.4 Related work in Sindhi Language

Rahman [39] discuss the Noun inflectionof Sindhi Language.Mahar et.al[25] worked on Sindhi word Segmentation. For Sindhi word prediction a bi- gram tri-gram and four-gram statistical model was developed by Mahar [26]. Internal structure of Sindhi was described by Lashari[21].

The first morphological analyzer for Sindhi Language was developed by Motlani[29]. Waquar[31] describe a Sindhi compound and complex word segmentation algorithm. First stemmer for Sindhi Language inPerso- Arabic Script was developed by Shah [44] by suffix stripping approach. Dootio[6] developed 2 algorithms for automatic stemming and Lemmatization of Sindhi Language in Parso Arabic Script. Nathani, B et.al [32] developed an inflectional stemmer for Sindhi Language in Devanagari Scriptand got 85% accuracy. A rule base lemmatizer [33] was developed by Nathani, B et.al, tested on 1000 words which gave 90% accuracy.

3. Methodology

In Indian Languages two main stemming approaches are reported (i) Rule based (ii) Machine Learning approach [43]. In Rule based approach manually handcrafted morphological rules are generated, which is a time-consuming process but having high accuracy. Machine learning approach further classified as Supervised and Unsupervised based on the type of data used for learning morphology. In supervised approach we need annotated data, which is not available for Sindhi Language. Rule based and Supervised Machine Learning approach both needs language expertise.

Unsupervised learning is defined as "the learning of morphology of a language by a computer system without any knowledge of language or language expertise". The character set of Sindhi Language is consisting of 38 consonants and 12 vowels, which makes this language morphological rich and highly inflected. It is very difficult and time-consuming task to derive all morphological rules manually for highly inflected Language such as Sindhi. The unsupervised approach will give better results as compared to rule-based approach for highly inflected languages.

For Unsupervised learning of morphology, we have used a tool "Linguistica 5" [22] which is based on the Minimum description Length (MDL) algorithm. We have used the Command Line Interface of Linguistica with the default parameters such as Maximum affix length =4, Minimum Stem length=4 and Minimum Signature count i.e. minimum number of stems for a valid signature is 5. The Input to Linguistica is a raw corpus of target language and output is List of Signature. Where Signature define as stem and pair of suffixes. A stem can belong to a single signature i.e. a word can have one signature. It is an iterative algorithm. In every iteration the algorithm word is segmented to create a signature and check whether this new signature will reduce the corpus length and signature will be accepted when it is having minimum 5 stem. The algorithm will stop when it found no new accepted modification in signature or in segment.

4. Experimental Setup

4.1 Data

Sindhi is a resource poor language and very less amount of data is available on web, so our first task is to generate the training data in Unicode, first we have translated 10000 Hindi sentences into Sindhi Language in Devanagari Script which is domain independent. Statistics of Training data set is shown in table 1. For testing we have extracted 1000 words from Sindhi -Hindi Dictionary.

Table 1. Statistics of Training Data Set

Number of sentences	Number of Word type	Total token
10000	14084	182930

4.2 Algorithm

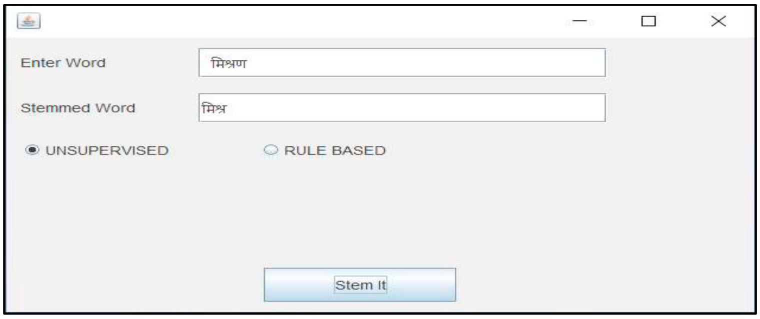
The following fig shown the algorithm for unsupervised stemming.

```
Input: Inflected Sindhi Word in Devanagari Script
Output: Stemmed Word (Stem)
Step1:    Input to “Linguistica 5” is 10000 sentences of Sindhi Language, in Devanagari Script.
          Output = List of Signature, Stems and affixes. Which is shown in Table:3.
Step 2: Collect all affixes and arrange them in decreasing order of length.
Step 3: Take input word i.e. inflected word.
Step 4: Match the input word with list of suffixes. If the match is found go to step 5 else go to step 6.
Step 5: Remove the suffix from the inflected word and save this word as root word.
Step 6: Return the root word.
```

Fig. 1. Algorithm for unsupervised stemming of Sindhi Language.

4.3 Implementation

For extraction of suffixes “Linguistica 5” was used. Suffixes stripping rules was implemented in java, using net beans IDE 8.1. Following Figure 2 shows the sample output of this system.



Enter Word मिश्रण

Stemmed Word मिश्र

● UNSUPERVISED ○ RULE BASED

Fig. 2 Sample Output

5. Result and Discussion

In Literature various researchers proposed different methods of stemmer evaluation. Broadly they are divided in two categories i.e. direct and indirect.1. direct in this we study the output or compare the output with external data (i.e. Gold Standard) and 2. Indirect in which we measure the performance of NLP applications (such as IR, Machine Translation) in which segmented word is used. Indirect approach is time consuming. Direct measures can perform in two ways first the results are evaluated by the language experts, and the second way are the results are compared with linguistic reference (i.e. gold Standard).

To evaluate our system first we have used direct approach. We evaluate our results by a language expert. Sample output of “Linguistica5” [22] is shown in Table 2. For evaluation we have created a test set of 1000 words, taken from a Sindhi Dictionary. Accuracy is calculated using following formula and got 87 % accuracy which is greater than existing previous Rule Stemmer based Stemmer[32] but less than rule based Lemmatizer [33].Comparative analysis is shown in Figure 3.

$$\text{Accuracy} = \frac{Correctaly\ stemed\ word}{Total\ stemmed\ word} \tag{1}$$

Table 2 Results of Linguistica 5

Total Number of Stems	Total Signatures	Total affix
2197	83	44

Table 3 Sample output of Linguistica 5

Total Number of Stems	Total Signatures
NULL/ूं	उलझन,ख़ोराक,ज़ाल, शिकासलाह
NULL/मंद	जरूरत, प़फ़ाइदे, फाइदे, फाय, फिक्र, भरोसे, सेहत
NULL/ऊं	अव्यवस्था, कणिका, शल्यक्रिया, संभावना, संवेदना, सुविधा
NULL/युनि/यूं	औषधि, गतिविधि, गाल्हि, जाति, नीति, पद्धति, रांदि, शक्ति

5.1 Stemming Error

There are two types of error occurs during the stemming process i.e. under stemming and over stemming[35]. Under stemming is when all closely related words not stemmed to same stem. Over stemming occurs when words of different category stemmed to same stem. Although the system gives adequate accuracy but it also gives some over stemming and under stemming errors. Table 4 and 5 shows some instances of under stemming and over stemming errors generated by system.

Table 4 under Stemming Errors Example

List of Inflected Words	Output Stem	Actual Stem
समस्याउनि,समस्याऊं	समस्य ,समस्या	समस्या
संस्थाऊं,संस्थाननि	संस्था, संस्थान	संस्था
संक्रामक,संक्रमण	संक्राम,संक्रम	संक्रमण

Table 5 Over Stemming Errors Example

List of Inflected Words	Output Stem	Actual Stem
मिश्रणमिश्रीअ	मिश्र	मिश्रण, मिश्री
मस्तिक,मस्ती	मस्त	मस्तिक, मस्त
मच्छरमच्छी	मच्छ	मच्छरमच्छी

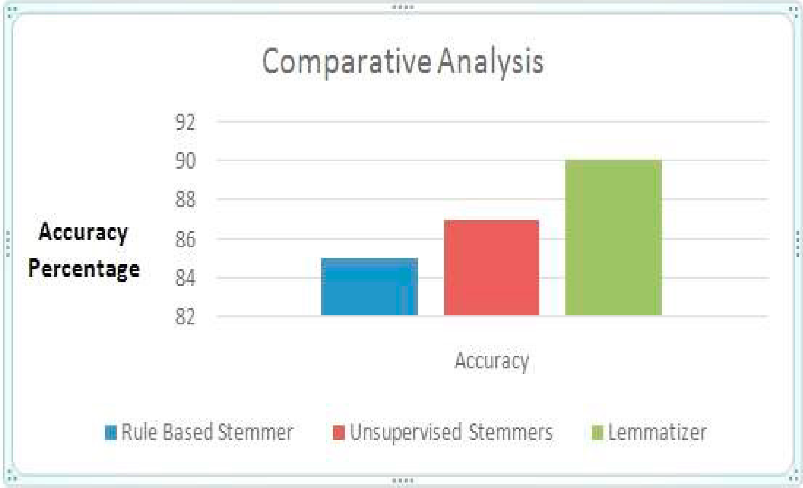


Fig :3 Comparative Analysis

6. Conclusion and Future Work

The proposed unsupervised stemmer provided increased accuracy. While rule based stemmer gave 85% accuracy [32], it gave 87% accuracy. Further the performance of proposed stemmer was evaluated using Direct approach,however, later on evaluation can be done in terms of performance of application of stemmed words in IR and Machine Translation.

For unsupervised stemming various algorithms are available in literature, in this work MDL based algorithm has been used. In future this work will be carried out using some other unsupervised algorithm and to improve accuracy i.e. to reduce over stemming and under stemming errors, we can extend this algorithm by applying some lemmatization rules, as existing Lemmatizer[33] gave 90%.

References

[1] Al-Omari, A., &Abuata, B. (2014). Arabic light stemmer (ARS). Journal of Engineering Science and Technology, 9(6), 702-717.

[2] Bharati, A., Sangal, R., Bendre, S., Kumar, P., & Aishwarya, K. R. (2001, November). Unsupervised Improvement of Morphological Analyzer for Inflectionally Rich Languages. In NLPRS (pp. 685-692).

[3] Bhat, S. (2012). Morpheme segmentation for kannada standing on the shoulder of giants. In Proceedings of the 3rd Workshop on South and Southeast Asian Natural Language Processing (pp. 79-94).

[4] Can Buğlalılar, B. (2017). UNSUPERVISED JOINT PART-OF-SPEECH TAGGING AND STEMMING FOR AGGLUTINATIVE LANGUAGES (Master's thesis, Fen BilimleriEnstitüsü).

[5] Can, B., &Manandhar, S. (2014, April). Methods and algorithms for unsupervised learning of morphology. In International Conference on Intelligent Text Processing and Computational Linguistics (pp. 177-205). Springer, Berlin, Heidelberg.

[6] Dootio, Mazhar &Wagan, Asim. (2017). AUTOMATIC STEMMING AND LEMMATIZATION PROCESS FOR SINDHI TEXT. 6. 19-28.

[7] Goldsmith, J. (2001). Unsupervised learning of the morphology of a natural language. Computational linguistics, 27(2), 153-198.

[8] Goldsmith, J. (2006). An algorithm for the unsupervised learning of morphology. Natural language engineering, 12(4), 353-371.

[9] Govilkar, S. S., Bakal, J. W., & Kulkarni, S. R. (2016). Extraction of Root Words using Morphological Analyzer for Devanagari Script. International Journal of Information Technology and Computer Science (IJITCS), 8(1), 33.

[10] Gupta, D., Kumar, Y. R., &Sajan, N. (2012). Improving unsupervised stemming by using partial lemmatization coupled with data-based heuristics for Hindi. International Journal of Computer Applications, 38(8), 1-8.

[11] Gupta, V., Joshi, N., & Mathur, I. (2013, September). Rule based stemmer in Urdu. In Computer and Communication Technology (ICCCT), 2013 4th International Conference on(pp. 129-132). IEEE.

[12] Gupta, V., Joshi, N., & Mathur, I. (2015, February). Design & development of rule based inflectional and derivational Urdu stemmer 'Usal'. In Futuristic Trends on Computational Analysis and Knowledge Management (ABLAZE), 2015 International Conference on (pp. 7-12). IEEE.

[13] Hammarström, H. (2009). Unsupervised Learning of Morphology and the Languages of the World.

[14] Hammarström, H. (2011). Radboud Universiteit and Max Planck Institute for Evolutionary Anthropology. Lars Borin, University of Gothenburg: Survey Article-Unsupervised Learning of Morphology,(c).

[15] Husain, M. S. (2012). An unsupervised approach to develop stemmer. International Journal on Natural Language Computing (IJNLC), 1(2), 15-23.

[16] Jivani, A. G. (2011). A comparative study of stemming algorithms. Int. J. Comp. Tech. Appl, 2(6), 1930-1938.

[17] Kanuparthi, N., Inumella, A., & Sharma, D. M. (2012, June). Hindi derivational morphological analyzer. In Proceedings of the Twelfth Meeting of the Special Interest Group on Computational Morphology and Phonology (pp. 10-16). Association for Computational Linguistics.

[18] Karanikolas, N. N. (2014). A Methodology for Building Simple but Robust Stemmers without Language Knowledge: Stemmer Configuration. Procedia-Social and Behavioral Sciences, 147, 370-375.

[19] Khan, S. A., Anwar, W., Bajwa, U. I., & Wang, X. (2012, December). A light weight stemmer for Urdu language: a scarce resourced language. In 24th international conference on computational linguistics (p. 69).

[20] Krishn, A., Guha, R. S., & Mukherjee, A. (2012). Unsupervised Morphological Analysis of Hindi.

[21] Lashari, M. A., & Soomro, A. A. (2013). Subject-Verb Agreement in Sindhi and English: A Comparative Study. Language in India, 13(6), 473-495.

[22] Lee, J., & Goldsmith, J. (2016). Linguistica 5: Unsupervised learning of linguistic structure. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations (pp. 22-26).

[23] Lovins, J. B. (1968). Development of a stemming algorithm. *Mech. Translat. & Comp. Linguistics*, *11*(1-2), 22-31.–31. 1977

[24] Lushtak, S. A. (2013). Unsupervised Morphological Word Clustering (Doctoral dissertation).

[25] Mahar, J. A., &Memon, G. Q. (2011). Probabilistic Analysis of Sindhi Word Prediction using N-Grams. Australian Journal of Basic and Applied Sciences, 5(5), 1137-1143

[26] Mahar, J. A., Memon, G. Q., &Danwar, S. H. (2011). Algorithms for Sindhi Word Segmentation using Lexicon-Driven Approach. International Journal of Academic Research, 3(3).

[27] Majgaonker, M. M., & Siddiqui, T. J. Discovering suffixes: A Case Study for Marathi.

[28] Makhija, S. D. (2016, March). A Study of Different Stemmer for Sindhi Language Based on Devanagari Script. In Computing for Sustainable Global Development (INDIACom), 2016 3rd International Conference on (pp. 2326-2329). IEEE

[29] Motlani, R., Tyers, F. M., & Sharma, D. M. (2016). A Finite-State Morphological Analyser for Sindhi. In LREC.

[30] Narejo, W. A., & Mahar, J. A. (2016, April). Morphology: Sindhi Morphological Analysis for Natural Language Processing Applications. In Computing, Electronic and Electrical Engineering (ICE Cube), 2016 International Conference on (pp. 27-31). IEEE.

[31] Narejo, W. A., Mahar, J. A., Mahar, S. A., Surahio, F. A., &Jumani, A. K. (2016). Sindhi Morphological Analysis: An Algorithm for Sindhi Word Segmentation into Morphemes. International Journal of Computer Science and Information Security, 14(6), 293.

[32] Nathani, B., Joshi, N., & Purohit, G. N. (2018, November). 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) (pp. 1-4). IEEE.

[33] Nathani, B., Joshi, N., & Purohit, G. N. (2019). Design and development of lemmatizer for Sindhi language in devanagri script. *Journal of Statistics and Management Systems*, *22*(4), 635-641.

[34] Oad, J. D. (2012). Implementing GF Resource Grammar for Sindhi Language (Doctoral dissertation, M.Sc. thesis, Chalmers University of Technology, Gothenburg, Sweden).

[35] Paice, C. D. (1994, August). An evaluation method for stemming algorithms. In Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval (pp. 42-50). Springer-Verlag New York, Inc..

[36] Pandey, A. K., & Siddiqui, T. J. (2008, July). An unsupervised Hindi stemmer with heuristic improvements. In Proceedings of the second workshop on Analytics for noisy unstructured text data (pp. 99-105). ACM.

[37] Patel, M., & Shah, A. (2016). An unsupervised stemming: A review. International Journal of Computer Science and Information Security, 14(7), 476.

[38] Porter, M. (1980). "An algorithm for suffix strippingprogram", Vol. 14, pp. 130-137

[39] Rahman, M. U. (2009). Sindhi Morphology and Noun Inflections. In Proceedings of the Conference on Language & Technology (pp. 74-81).

[40] Ramanathan, A., & Rao, D. D. (2003, April). A lightweight stemmer for Hindi. In *the Proceedings of EACL*.

[41] Saharia, N., Konwar, K. M., Sharma, U., &Kalita, J. K. (2013, March). An improved stemming approach using HMM for a highly inflectional language. In International Conference on Intelligent Text Processing and Computational Linguistics (pp. 164-173). Springer, Berlin, Heidelberg.

[42] Saharia, N., Sharma, U., &Kalita, J. (2012, August). Analysis and evaluation of stemming algorithms: a case study with Assamese. In Proceedings of the International Conference on Advances in Computing, Communications and Informatics (pp. 842-846). ACM.

[43] Saharia, N., Sharma, U., &Kalita, J. (2014). Stemming resource-poor Indian languages. ACM Transactions on Asian Language Information Processing (TALIP), 13(3), 14.

[44] Shah, M., Shaikh, H., Mahar, J., & Mahar, S. (2016). Sindhi Stemmer for Information Retrieval System using Rule-Based Stripping Approach. Sindh University Research Journal-SURJ (Science Series), 48(4).

[45] Suba, K., Jiandani, D., & Bhattacharyya, P. (2011). Hybrid inflectional stemmer and rule-based derivational stemmer for gujarati. In Proceedings of the 2nd Workshop on South Southeast Asian Natural Language Processing (WSSANLP)(pp. 1-8).

[46] Virpioja, S., Turunen, V. T., Spiegler, S., Kohonen, O., &Kurimo, M. (2011). Empirical Comparison of Evaluation Methods for Unsupervised Learning of Morphology. TAL, 52(2), 45-90