A Rule-Based Stemmer for Punjabi Verbs

Prabhjot Kaur[1], Preetpal Kaur Buttar[2]

[1]M. Tech. Research Scholar, Department of Computer Science and Engineering, Sant Longowal Institute of Engineering and Technology, Longowal, Sangrur - 148106, Punjab, India
[2]Assistant Professor, Department of Computer Science and Engineering, Sant Longowal Institute of Engineering and Technology, Longowal, Sangrur - 148106, Punjab, India

---***---

Abstract – *This paper proposes a rule-based stemmer for the stemming of verbs in Punjabi language. The proposed Punjabi Verb Stemmer (PVS) uses a rule-based approach for the stemming of Punjabi verbs. A database containing valid root verbs occurring in the Punjabi language has been created. The database stores 3,135 Punjabi root verbs. When a verb is input to the system, it is first compared with the database of root verbs to discover whether the input verb is a root verb or an inflected verb. If the input verb is a root verb, no stemming is required and the input verb is returned as the output. Otherwise, the input verb (inflected) is sent to the suffix stripping algorithm to get the corresponding root verb which uses a predefined suffix list. Punjabi is a resource-scarce language with a very few linguistic resources developed so far. In case of stemming in Punjabi language, most of the work done so far has concentrated on stemming of noun words and proper names, no published work for the stemming of Punjabi verbs has been reported. The proposed PVS has an overall accuracy of 95.21%. This stemmer can contribute to many applications of natural language processing and text mining.*

Key Words: Stemming, Stemmer, Suffix Stripping, Verbs, Punjabi

1. INTRODUCTION

Stemming is a pre-processing task in the information retrieval systems [1], which reduces an inflected word to its stem, root or base form [2, 3]. For example, a stemming algorithm reduces the words “discussion”, “discussing” and “discussed” to the stem “discuss”. Stemming is used in the information retrieval tasks to improve system performance by enhancing the ability to match document queries [3]. For example, when a user enters a query word ‘applying’, s/he may also want to retrieve the documents that contain the word ‘apply’ and ‘applied’ as well [4]. Stemming is used to improve the efficiency of various search engines to get the results [5].

Various methods have been used for stemming such as stochastic algorithm, brute force algorithm, suffix stripping algorithm, n-gram analysis, lemmatization algorithm, hybrid approaches, etc. These methods differ in terms of accuracy and performance [3]. Ramanathan and Rao [6] proposed a lightweight stemmer for Hindi language. Their proposed stemmer conflates the similar words using suffix removal. The suffix lists for the stemmer were created manually. Dasgupta and Ng [7] proposed an algorithm for the unsupervised morphological parsing of Bengali language by segmenting the words into prefixes, suffixes, and stems. Pandey and Siddiqui [8] suggested an unsupervised stemming algorithm for Hindi language. The algorithm was based on division technique. Zehurul et.al [9] proposed a lightweight stemmer for Bengali language based on a predefined suffix list which removes the suffix of a word on a longest-match basis. Suba et.al [10] suggested two kinds of stemmers for Gujarati language – a lightweight inflectional stemmer based on a hybrid approach and heavyweight derivational stemmer based on a rule-based approach. Kumar and Rana [3] proposed a stemmer for Punjabi language based on the duo of brute force and suffix stripping approaches. Gupta and Lehal [11] proposed a stemmer for noun and proper names occurring in Punjabi language. The stemmer obtained a stem or radix of Punjabi word and then searched in Punjabi noun and proper name dictionary. Mishra and Chandra [12] suggested MAULIK: a stemmer for Hindi language. The MAULIK algorithm was used to stem the Hindi words by using hybrid technique. Gupta [13] proposed a stemmer for verbs in Hindi language based on suffix-stripping. The stemmer applied a technique based on suffix removal rules in order to stem Hindi verbs. Punjabi is a resource-scarce language with a very few linguistic resources developed so far. In case of stemming in Punjabi language, most of the work done so far has concentrated on stemming of noun words and proper names, no stemming is required and the input verb is returned as the output. In this paper, a rule-based approach has been proposed for the stemming of Punjabi verbs. Our approach is similar to those of Ramanathan and Rao [6] for Hindi and Zehurul et.al [9] for Bengali in which the suffix is stripped from the input verb using a predefined list on the basis of longest-match. PVS has an overall accuracy of 95.21%.

2. PROPOSED METHODOLOGY FOR STEMMING OF PUNJABI VERBS

2.1 Categorization of verbs in Punjabi language

A verb is intended to explain states, activities or events and the main components of predicates that form particular sentences [13]. In the Punjabi language, verbs can be categorized as one-word verbs or two-word verbs. For example, ਤੁਰਨਾ is a one-word verb with ਤੁਰ as root and ਨਾ as suffix. Similarly, ਇਕੱਠਾ ਕਰਨਾ is a two-word verb with ਇਕੱਠਾ ਕਰ as root and ਨਾ as suffix.

2.2 Approach used

In this paper, a rule-based approach has been proposed for the stemming of Punjabi verbs. A database containing valid root verbs existing in the Punjabi language has been created. The database stores 3,135 Punjabi root verbs. When a verb is input to the system, it is compared with the database of root verbs to check whether the input verb is a root verb or an inflected verb. If it is a root verb, no stemming is required and the input verb is returned as the output. Otherwise, the input verb (inflected) is sent to the suffix-stripping algorithm to get the corresponding root verb. This algorithm strips the suffix from the input verb on a longest-match basis. The figure below presents a schematic diagram of PVS.

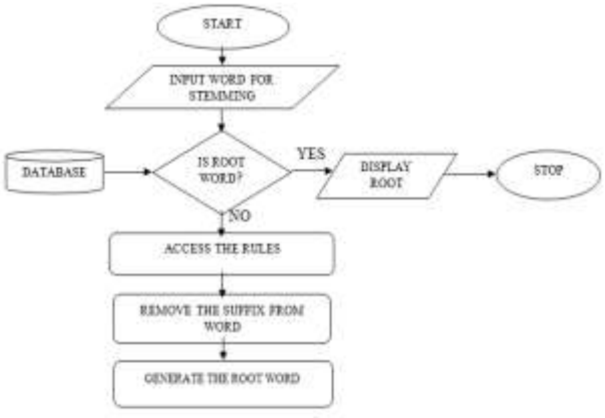

Fig -1: Schematic diagram of PVS

2.3 List of suffixes for Punjabi verbs

First, a total of 48 suffixes existing for Punjabi verbs were identified. Then, these suffixes were categorized into different lists on the basis of suffix length. These lists help to construct the rules for PVS. The lists of suffixes that we have used in stemmer are as below:

Table- 1: Suffix-list-1 having suffix length-6

Sr. No.	Suffix	Sr. No.	Suffix
1.	ਵਾਂਗੀਆ	4.	◌ਾਵਾਂਗਾ
2.	◌ਾਵਾਂਗੀ	5.	ਉਂਦੀਆਂ
3.	◌ਾਵਾਂਗੀ		

Table- 2: Suffix-list-2 having suffix length-5

Sr. No.	Suffix	Sr. No.	Suffix
1.	ਉਣੀਆਂ	3.	ਵਾਂਗਾ
2.	ਵਾਂਗੇ	4.	ਵਾਂਗੀ

Table- 3: Suffix-list-3 having suffix length-4

Sr. No.	Suffix	Sr. No.	Suffix
1.	ਉਣਾ	7.	◌ਂਦਾ
2.	ਉਣੀ	8.	◌ੰਦਾ
3.	ਉਣੇ	9.	◌ੰਦੀ
4.	ਓਗੇ	10.	◌ੰਦੇ
5.	◌ਂਦੇ	11.	◌ੀਦਾ
6.	◌ਂਦੀ		

Table- 4: Suffix-list-4 having suffix length-3

Sr. No.	Suffix	Sr. No.	Suffix
1.	ਦੀਆਂ	8.	◌ਾ◌ਂਗਾ
2.	ਨੀਆਂ	9.	◌ਾ◌ਂਗੀ
3.	ਉਂਦਾ	10.	◌ਾ◌ਂਗੇ
4.	ਣੀਆਂ	11.	ਵੇਗਾ
5.	ਉਂਦੇ	12.	◌ਂਦੀਆ
6.	ਉਂਦੀ	13.	ਵੇਗੀ
7.	ਵੋਗੇ		

Table- 5: Suffix-list-5 having suffix length-2

Sr. No.	Suffix	Sr. No.	Suffix
1.	ਦਾ	8.	ਣਾ
2.	ਦੀ	9.	ਣੇ
3.	ਦੇ	10.	ਨਾ
4.	ਇਆ	11.	ਨੇ
5.	ਦਿਆ	12.	ਨੀ
6.	ਉਣ	13.	ਈਏ
7.	ਣੀ	14.	◌ਿਆ

Table- 6: Suffix-list-6 having suffix length-1

Sr. No.	Suffix
1.	ਏ

2.4 Proposed rules for PVS

After generating the suffix lists, we have created rules for stemming based on the length of suffixes. We have made 48 rules for removal of suffix from the input verb. The most appropriate rule is matched with the input verb, and the suffix is stripped. The rules used for PVS are as follows:

Rule 1: if a word of Punjabi verb ends with ‘ਦਾ’, remove the suffix ‘ਦਾ’ at the end of word.

Rule 2: if a word of Punjabi verb ends with ‘ਦੀ’, remove the suffix ‘ਦੀ’ at the end of word.

Rule 3: if a word of Punjabi verb ends with ‘ਦੇ’, remove the suffix ‘ਦੇ’ at the end of word.

Rule 4: if a word of Punjabi verb ends with ‘ਨਾ, remove the suffix ‘ਨਾ’ at the end of word.

Rule 5: if a word of Punjabi verb ends with ‘ਨੀ’, remove the suffix ‘ਨੀ’ at the end of word.

Rule 6: if a word of Punjabi verb ends with ‘ਨੇ’, remove the suffix ‘ਨੇ’ at the end of word.

Rule 7: if a word of Punjabi verb ends with ‘ਦੀਆਂ’, remove the suffix ‘ਦੀਆਂ’ at the end of word.

Rule 8: if a word of Punjabi verb ends with ‘ਨੀਆਂ’, remove the suffix ‘ਨੀਆਂ’ at the end of word.

Rule 9: if a word of Punjabi verb ends with ‘ਈਆਂ’, remove the suffix ‘ਈਆਂ’ at the end of word.

Rule 10: if a word of Punjabi verb ends with ‘ਉਂਦਾ’, remove the suffix ‘ਉਂਦਾ’ at the end of word.

Rule 11: if a word of Punjabi verb ends with ‘ਉਂਦੀ’, remove the suffix ‘ਉਂਦੀ’ at the end of word.

Rule 12: if a word of Punjabi verb ends with ‘ਉਣਾ’, remove the suffix ‘ਉਣਾ’ at the end of word.

Rule 13: if a word of Punjabi verb ends with ‘ਉਣੀ’, remove the suffix ‘ਉਣੀ’ at the end of word.

Rule 14: if a word of Punjabi verb ends with ‘ਉਂਦੇ’, remove the suffix ‘ਉਂਦੇ’ at the end of the word.

Rule 15: if a word of Punjabi verb ends with ‘ਉਣੇ’, remove the suffix ਉਣੇ ‘’ at the end of the word.

Rule 16: if a word of Punjabi verb ends with ‘ਓਗੇ’, remove the suffix ‘ਓਗੇ’ at the end of word.

Rule 17: if a word of Punjabi verb ends with ‘ਾਂਗਾ’, remove the suffix ‘ਾਂਗਾ’ at the end of word.

Rule 18: if a word of Punjabi verb ends with ‘ਵਾਂਗੀ’, remove the suffix ‘ਵਾਂਗੀ’ at the end of word.

Rule 19: if a word of Punjabi verb ends with ‘ਵਾਂਗਾ’, remove the suffix ‘ਵਾਂਗਾ’ at the end of word.

Rule 20: if a word of Punjabi verb ends with ‘ਵਾਂਗੇ’, remove the suffix ‘ਵਾਂਗੇ’ at the end of word.

Rule 21: if a word of Punjabi verb ends with ‘ਉਣੀਆਂ’, remove the suffix ‘ਉਣੀਆਂ’ at the end of word.

Rule 22: if a word of Punjabi verb ends with ‘ਉਂਦੀਆਂ’, remove the suffix ‘ਉਂਦੀਆਂ’ at the end of word.

Rule 23: if a word of Punjabi verb ends with ‘ਾਵਾਂਗੇ’, remove the suffix ‘ਾਵਾਂਗੇ’ at the end of word.

Rule 24: if a word of Punjabi verb ends with ‘ਾਵਾਂਗਾ’, remove the suffix ‘ਾਵਾਂਗਾ’ at the end of word.

Rule 25: if a word of Punjabi verb ends with ‘ਵਾਂਗੀਆ’, remove the suffix ‘ਵਾਂਗੀਆ’ at the end of word.

Rule 26: if a word of Punjabi verb ends with ‘ਾਂਗੀ’, remove the suffix ‘ਾਂਗੀ’ at the end of word.

Rule 27: if a word of Punjabi verb ends with ‘ਾਂਗੇ’, remove the suffix ‘ਾਂਗੇ’ at the end of word.

Rule 28: if a word of Punjabi verb ends with ‘ਵੇਗਾ’, remove the suffix ‘ਵੇਗਾ’ at the end of word.

Rule 29: if a word of Punjabi verb ends with ‘ਂਦੀਆ’, remove the suffix ‘ਂਦੀਆ’ at the end of word.

Rule 30: if a word of Punjabi verb ends with ‘ਵੇਗੀ’, remove the suffix ‘ਵੇਗੀ’ at the end of word.

Rule 31: if a word of Punjabi verb ends with ‘ਵੋਗੇ’, remove the suffix ‘ਵੋਗੇ’ at the end of word.

Rule 32: if a word of Punjabi verb ends with ‘ਂਦੇ’, remove the suffix ‘ਂਦੇ’ at the end of word.

Rule 33: if a word of Punjabi verb ends with ‘ਂਦਾ’, remove the suffix ‘ਂਦਾ’ at the end of word.

Rule 34: if a word of Punjabi verb ends with ‘ੰਦਾ’, remove the suffix ‘ੰਦਾ’ at the end of word.

Rule 35: if a word of Punjabi verb ends with 'ੰਦੀ', remove the suffix 'ੰਦੀ' at the end of word.

Rule 36: if a word of Punjabi verb ends with 'ੰਦੇ', remove the suffix 'ੰਦੇ' at the end of word.

Rule 37: if a word of Punjabi verb ends with 'ੀਦਾ', remove the suffix 'ੀਦਾ' at the end of word.

Rule 38: if a word of Punjabi verb ends with 'ਇਆ', remove the suffix 'ਇਆ' at the end of word.

Rule 39: if a word of Punjabi verb ends with 'ਦਿਆ', remove the suffix 'ਦਿਆ' at the end of word.

Rule 40: if a word of Punjabi verb ends with 'ਉਣ', remove the suffix 'ਉਣ' at the end of word.

Rule 41: if a word of Punjabi verb ends with 'ਣੀ', remove the suffix 'ਣੀ' at the end of word.

Rule 42: if a word of Punjabi verb ends with 'ਣਾ', remove the suffix 'ਣਾ' at the end of word.

Rule 43: if a word of Punjabi verb ends with 'ਣੇ', remove the suffix 'ਣੇ' at the end of word.

Rule 44: if a word of Punjabi verb ends with 'ਈਏ', remove the suffix 'ਈਏ' at the end of word.

Rule 45: if a word of Punjabi verb ends with 'ਿਆ', remove the suffix 'ਿਆ' at the end of word.

Rule 46: if a word of Punjabi verb ends with 'ਾਵਾਂਗੀ', remove the suffix 'ਾਵਾਂਗੀ' at the end of word.

Rule 47: if a word of Punjabi verb ends with 'ਂਦੀ', remove the suffix 'ਂਦੀ' at the end of word.

Rule 48: if a word of Punjabi verb ends with 'ਏ', remove the suffix 'ਏ' at the end of word.

2.5 Proposed Algorithm

```
Step 1: Read word from user input containing only verbs.
Step 2: Check whether the entered word is root word.
        If the entered word is root word, then print the word
        as output.
        Go to step 1.
Step 3: SUFF = Last six characters from the right side of the
        word.
        If SUFF exists in the suffix-list-1 (length-6), then:
                Print the root word obtained after stripping
                SUFF, go to step 1.
Step 4: SUFF = Last five characters from the right side of the
        word.
        If SUFF exists in the suffix-list-2 (length-5), then:
                Print the root word obtained after stripping
                SUFF, go to step 1.
Step 5: SUFF = Last four characters from the right side of the
        word.
        If SUFF exists in the suffix-list-3 (length-4), then:
                Print the root word obtained after stripping
                SUFF, go to step 1.
Step 6: SUFF = Last three characters from the right side of
        the word.
        If SUFF exists in the suffix-list-4 (length-3), then:
                Print the root word obtained after stripping
                SUFF, go to step 1.
Step 7: SUFF = Last two characters from the right side of the
        word.
        If SUFF exists in the suffix-list-5 (length-2), then:
                Print the root word obtained after stripping
                SUFF, go to step 1.
Step 8: SUFF = Last one character from the right side of the
        word.
        If SUFF exists in the suffix-list-6 (length-1), then:
                Print the root word obtained after stripping
                SUFF, go to step 1.
Step 9: Exit.
```

At step 1, the word is read from the user input which should be a Punjabi verb. At step 2, the entered word is searched in the root verb database, to find whether the entered word is a root word or not. If the input word is not present in the database, then we go to step 3. At this step, the last six characters from the right side of the word are extracted. If the extracted characters exist in the suffix-list-1 (the list having length of suffix equal to 6), then the matching suffix is removed from the entered word. If the extracted characters do not exist in suffix-list-1, then suffix-list-2 is searched and so on until the last characters of the word do not match with the suffix list. If matched, then the suffix is removed from the word.

3. RESULTS AND DISCUSSION

Accuracy of the stemmer depends on the words in the database (i.e. the list of root words) and the rules created to remove suffixes. We have stored 3,135 words in the database. It decreases the probability of switching to the second option for removing suffixes [12].

To calculate the accuracy of the system, we input 3,135 words to PVS for analysis. Among these 3,135 words, 2,985 words have been correctly evaluated and 150 words are invalid because they violate specific and general rules. The accuracy of our stemmer is 95.21%.

4. CONCLUSION

Stemming produces linguistically standardized text that helps in enhancing the results of information retrieval tasks [1]. Our proposed stemmer works with Punjabi verbs which uses a rule-based suffix-stripping technique to perform the stemming of the Punjabi verbs. Punjabi is a resource-scarce language with stemmers existing for nouns and proper names only. PVS is the reportedly the first stemmer for the stemming of Punjabi verbs with an overall accuracy of 95.21%. PVS can be enhanced by enabling it for the stemming of Punjabi adverbs and adjectives. This stemmer can contribute to many applications of information retrieval and natural language processing.

REFERENCES

[1] R. Puri, R. P. S. Bedi, V. Goyal, "Punjabi stemmer using Punjabi WordNet database," Indian journal of science and technology, vol. 8 no. 27, Oct. 2015, pp. 1-5.

[2] G. Joshi, K. D. Garg, "Enhanced version of Punjabi stemmer using synset," Advanced Research in Computer Science and Software Engineering, vol. 4 no. 5, 2014.

[3] D. Kumar, P. Rana, "Design and development of a stemmer for Punjabi," International Journal of Computer Applications, vol. 11 no. 12, Dec. 2010, pp. 18-23.

[4] V. Gupta, G. S. Lehal, "A survey of common stemming techniques existing stemmers for Indian languages," Journal of Emerging Technologies in Web Intelligence, vol. 5 no. 2, pp. 157-161.

[5] P. Thapar, "A hybrid approach used to stem Punjabi words," International Journal of Computer Science and Mobile Computing, vol. 3 no. 11, 2014, pp. 1-9.

[6] A. Ramanathan, D. Rao, "A lightweight stemmer for Hindi," Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics, on Computational Linguistics for South Asian Languages, 2003, pp. 43-48.

[7] S. Dasgupta, V. Ng, "Unsupervised morphological parsing of Bengali," Language Resources and Evaluation, vol. 40, 2006, pp. 311-330.

[8] A. K. Pandey, T. J. Siddiqui, "An unsupervised Hindi stemmer with heuristic improvements," Proceedings of the Second Workshop on Analytics for Noisy Unstructured Text Data, 2008, pp. 99-105.

[9] M. Z. Islam, M. N. Uddin, M. Khan, "A light weight stemmer for Bengali and its use in spelling checker," Proceedings of the First International Conference on Digital Communication and Computer Applications (DCCA07), Irbid, Jordan, 2007.

[10] K. Suba, D. Jiandani, P. Bhattacharyya, "Hybrid inflectional stemmer and rule-based derivational stemmer for Gujarati," Proceedings of the 2nd Workshop on South and Southeast Asian Natural Language Processing (WSSANLP), IJCNLP 2011, ChiangMai, Thailand, 2011, pp. 1-8.

[11] V. Gupta, G. S. Lehal, "Punjabi language stemmer for nouns and proper names," Proceedings of the 2nd Workshop on South and Southeast Asian Natural Language Processing (WSSANLP), IJCNLP 2011, Chiang Mai, Thailand, 2011, pp. 35-39.

[12] U. Mishra, C. Prakash, "MAULIK: An effective stemmer for Hindi language," International Journal of Computer Science and Engineering, vol. 4, 2012; pp. 711–717.

[13] V. Gupta, "Suffix stripping based verb stemming for Hindi," International Journal of Advanced Research in Computer Science and Software Engineering, vol. 4 no. 1, 2014, pp. 179-181.