



## A RULE-BASED STEMMER FOR PUNJABI ADJECTIVES

Harmanjeet Kaur

Department of Computer Science and Engineering  
Sant Longowal Institute of Engineering and Technology  
Longowal, Sangrur, India

Preetpal Kaur Buttar

Department of Computer Science and Engineering  
Sant Longowal Institute of Engineering and Technology  
Longowal, Sangrur, India

**Abstract:** This research work is concerned with the development of a rule-based stemmer for stemming of adjectives in the Punjabi language. Stemming is a method of deriving the root word from the inflected word. The proposed Punjabi Adjective Stemmer (PAS) uses a rule-based approach for converting the inflected Punjabi adjectives to their root forms. A database containing valid root adjectives occurring in the Punjabi language has been created. This database stores 1,762 Punjabi root adjectives. When an adjective word is fed to PAS as an input, first it compares the input word with the root database to determine whether the input adjective is a root adjective or an inflected one. If the input adjective is a root adjective, then no stemming is required and the input adjective is returned as the output. Otherwise, the inflected input adjective is sent to the suffix-stripping algorithm to get the corresponding root adjective. The suffix-stripping algorithm uses a set of predefined rules. India is a linguistically rich country with 22 languages recognized officially. But the computational resources developed for these languages are very scarce. Most of the stemmers developed for Punjabi language so far concentrated on nouns and proper names. PAS is the only stemmer developed so far for specifically addressing the problem of stemming of Punjabi adjectives. PAS has an overall accuracy of 88.76%.

**Keywords:** Stemming, Stemmer, Suffix-Stripping, Adjectives, Punjabi

### I. INTRODUCTION

Stemming is the technique of removing affix of the input words (e.g., admitted, admitting) to their base form (e.g., admit). The “root” during this case might not be a true root word, however simply a canonical sort of the first word. Stemming uses a heuristic approach that cuts the end of a word to properly reforming words into its root form. Therefore, the words “admitted”, “admitting” would possibly be converted to “admit” instead of “admitting” as a result of the ends was simply shredded off.

There are different algorithms that can be used in the stemming process like Porter stemming algorithm, dictionary-based algorithm, rule-based algorithm, hybrid algorithm, stochastic algorithm, corpus-based algorithm, n-gram algorithm, matching algorithm etc. These all stemming techniques differ on the basis of their performance, accuracy and how to handle all the stemming interruptions [1].

### II. LITERATURE REVIEW

A. Sharma et al. [2] proposed a stemming algorithm for Hindi information retrieval that uses a hybrid approach (a combination of brute force approach, suffix-stripping approach, and suffix substitution). This stemming algorithm was implemented for Hindi noun words and gives an accuracy of 92.2%.

D. Kumar et al. [3] suggested stemming of the Punjabi words using brute force and suffix-stripping techniques to get maximum result. They got 81.27% accuracy with the database of 52000 words.

N. Saharia et al. [4] adopted the suffix removal approach alongside a rule that developed all the possible suffixes. They

got 82% accuracy by using the suffix removal approach with 20,000 root-word list.

D. Bijal et al. [5] proposed an outline of stemming algorithms for Indian and non-Indian languages and recommended that stemming will considerably increase the retrieval results for each rule-based and statistical approach. Jivani et al. [6] suggested different methods of stemming and their comparison on the basis of usage, advantages and disadvantages. In this paper, they categorized stemming in three groups: truncating, statistical and mixed.

Pandey et al. [7] proposed a rule-based stemmer of Marathi WordNet with the help of stem exception dataset and named entities. In this paper, they used Marathi WordNet to reduce the issue of over-stemming and under-stemming. They developed a hybrid system using a rule-based and machine learning approach to make the system more perfect.

I. Slawik et al. [8] proposed integration techniques for selective stemming in SMT systems for removal of adjectives. In their paper, they conferred a technique to attenuate the issues of data sparsity while translating morphologically rich languages into less inflected languages for stemming of some word types.

V. Gupta [9] proposed an automatic stemming of words for Punjabi language with eighteen suffixes for Punjabi nouns and proper names and sort of completely different suffixes for the Punjabi verbs, adjectives and adverbs using totally different stemming rules.

R. Puri et al. [10] developed a Punjabi WordNet database for the Punjabi stemmer. In this paper, suffix-stripping approach with set of suffix removal rules has been used for creating a Punjabi stemmer. The algorithmic program mentioned in this paper uses regular expressions for locating suffix matches.

A. Ramanathan et al. [11] proposed a lightweight stemmer for Hindi language. Suffix removal technique has been used to

stem nouns, Verbs and adjectives in Hindi language. The proposed system helps to overcome the problem of over-stemming and under-stemming.

H. Singh [12] proposed study on Punjabi stemmers that puts focus on brute force technique. The papers conclude that approaches used by different researchers are almost same there is no big difference and we need to make new path in this area.

D. Kumar *et al*. [13] suggested a method for design and development of Punjabi stemmer. In this paper, they are using brute force and suffix stripping technique and got 80.73% accuracy.

C. Dhawan *et al*. [14] are proposed Punjabi stemmer based on hybrid approach. They had used brute force, suffix substitution and suffix removal to overcome the problem of over-stemming and under-stemming.

V. Gupta *et al*. [15] proposed stemmer for the Urdu language by using the rule based technique. Stemmer is useful for complex and morphological rich words and they got 86.5% accuracy.

S. Paul *et al*. [16] suggested a design of rule-based Hindi lemmatizer. In this research, their aim was optimize time and obtain accurate results. They proposed 89.08% accuracy.

V. Gupta *et al*. [17] proposed survey of stemming techniques and existing stemmer of Indian language. Dasgupta *et al*. [18] proposed an unsupervised morphological analysis of Bengali language. The algorithm is used to segment words into stem, suffixes and prefixes with no prior knowledge of the morphological rules of a particular language. It calculated on a group of chunks 4-110 phonetic words of Bengali language, the algorithm has the F-score 83% degree, considerably superior to semantics, one of the most used unmonitored morphological analyst, with about 23%.

Goldsmith [19] developed an approach of unsupervised learning for languages morphology dependent on the “Minimum Description Length (MDL)”, that focus on presenting the foundations of words as compactly as possible. This study reports based on the analysis of MDL to simulate an uncontrolled study of the segmentation of European languages using enclosures ranging from 5,000 to 5,00,000 words. The result corresponds well to the analysis, which will be expressed by the human morphologist.

Singh *et al*. [20] suggested systematic reviews of text stemming techniques. In their paper, they described existing text stemming techniques by classifying them on the idea of some key parameters as entirely totally different assortment schemes, variant size, and nature of corpus, multiple tasks so on. They created public at the end of article that unsupervised stemming offers future directions to the researchers to enhance the performance of unsupervised corpus-based stemming ways.

### III. PROPOSED METHODOLOGY FOR STEMMING OF PUNJABI ADJECTIVES

#### A. Categorization of adjectives in Punjabi language

An adjective shows some of the features of a noun or pronoun. In the Punjabi language, an adjective usually comes before the nouns but follows the pronouns. Punjabi adjectives are categorized into two parts:

- Inflected adjectives- These adjectives operate as nouns and they change their form for gender, number

and case, e.g. ਸੋਹਣਾ sohna “handsome”, ਕਾਲਾ kala “black”.

- Uninflected adjectives- This class of adjectives do not change form for gender, number, or case, e.g. ਮਿਹਨਤੀ mehnatī ‘hardworking’, ਮਸ਼ਹੂਰ mashhūr ‘famous’, etc.

In this paper, the rule-based approach has been used for stemming of Punjabi adjectives. Moreover, this approach comes with a combination of brute force and suffix-stripping technique to make the system more efficient and perfect.

#### B. Approach used

In this paper, the rule-based approach has been used for stemming of Punjabi adjectives. We have created a database of 1,762 root adjective words. To collect database we have gone through Punjabi dictionaries and various other resources and try to cover commonly used words. We have referred many resources that are given below:

<http://dic.learnpunjabi.org/default.aspx>

<http://punjabi.aglsoft.com/punjabi/learngrammar/?show=adjective>

<https://www.enchantedlearning.com>

Punjabi Vocabulary of Common Terms by Punjabi University

Swan pocket English to Punjabi dictionary

English to Punjabi ShabdKosh

A Punjabi adjective word is entered as input to PAS. Then, PAS searches for the matching word in the root database to determine whether the input word is inflected or not. If the input adjective word is located in the root database, then it means that it is not inflected. In this case, PAS will provide the same word as output. Otherwise, the input word will be passes to the suffix-stripping algorithm to stem the suffix from the end of the input word using suffix-stripping rules defined in the system. The figure below represents the schematic diagram of PAS.

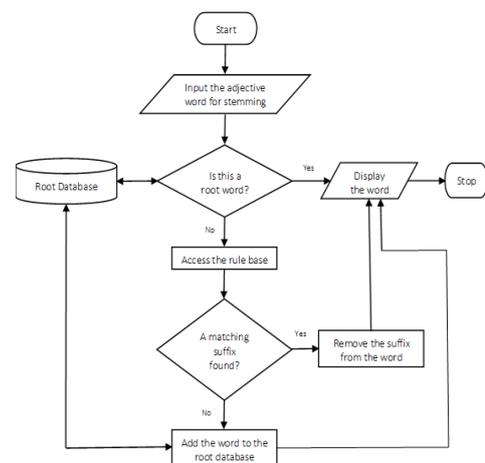


Figure1: Schematic diagram of PAS

#### C. List of Suffixes for Punjabi Adjectives

We have grouped up adjective suffixes into different lists on the basis of suffix length. These lists help to develop rules for PAS. After thorough analysis of the adjectives which exist in Punjabi language, we discovered that the suffixes for adjectives range from minimum 1 character to maximum 5

characters in length. Following are the 5 suffix lists according to the length of suffixes.

Tables 1: Suffix list-1 having suffix length-5

Sr.No	Suffix	Examples
1.	ਪ੍ਰਸ	ਅਰਾਮਪਰਸਤੀ, ਵਪਾਰਪਰਸਤੀ
2.	ਸ਼ੀਲਤਾ	ਉਤੇਜਿਤਸ਼ੀਲਤਾ, ਸਹਿਣਸ਼ੀਲਤਾ
3.	ਪੂਰਵਕ	ਆਗਿਆਪੂਰਵਕ,ਸ਼ਾਂਤਪੂਰਵਕ

Table 2: Suffix list-2 having suffix length-4

Sr.	Suffix	Examples
1.	ਪਾਤਰ	ਵਿਸ਼ਵਾਸਪਾਤਰ, ਦਾਨਪਾਤਰ
2.	ਪੂਰਣ	ਨਿਰਦੋਸ਼ਪੂਰਣ,ਦਿਆਲਪੂਰਣ
3.	ਪੁਣਾ	ਗੰਵਾਰਪੁਣਾ,ਆਲਸਪੁਣਾ
4.	ਬਾਜ਼	ਚਾਲਬਾਜ਼, ਧੋਖੇਬਾਜ਼,
5.	ਦਾਇਕ	ਅਰਾਮਦਾਇਕ, ਸੁੱਖਦਾਇਕ
6.	ਬਾਜ਼ੀ	ਚਾਲਬਾਜ਼ੀ, ਜਲਦਬਾਜ਼ੀ
7.	ਸ਼ਕਤੀ	ਉਪਜਾਉਸ਼ਕਤੀ, ਸਹਿਣਸ਼ਕਤੀ
8.	ਮੰਦੀ	ਅਕਲਮੰਦੀ, ਹਾਜ਼ਤਮੰਦੀ

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

Sr.	Suffix	Examples
1.	ਯੋਗ	ਸਹਿਣਯੋਗ, ਪ੍ਰਸੰਸਾਯੋਗ
2.	ਸ਼ੀਲ	ਉਤੇਜਿਤਸ਼ੀਲ, ਆਕਰਸ਼ਣਸ਼ੀਲ
3.	ਵਾਨ	ਗਿਆਨਵਾਨ, ਧਨਵਾਨ
4.	ਕਰਣ	ਸਪੱਸ਼ਟੀਕਰਣ,ਸੂਚੀਕਰਣ
5.	ਉਣਾ	ਪਿਲਪਿਲਾਉਣਾ, ਪੱਕਾਉਣਾ
6.	ਦਾਰ	ਗੁੰਝਲਦਾਰ, ਗੁੰਢੇਦਾਰ
7.	ਹੀਣ	ਗਿਆਨਹੀਣ, ਅਕਲਹੀਣ
8.	ਬੱਧ	ਵਚਨਬੱਧ
9.	ਘਾਤ	ਮਿੱਤਰਘਾਤ, ਵਿਸ਼ਵਾਸਘਾਤ
10.	ਪੋਸ਼	ਸਿਆਹਪੋਸ਼, ਸਫੇਦਪੋਸ਼
11.	ਮੰਦ	ਅਕਲਮੰਦ, ਸਿਹਤਮੰਦ
12.	ਖੋਰ	ਉਪਜਾਉਖੋਰ, ਗੁੱਸੇਖੋਰ
13.	ਕਾਰ	ਅਨੰਤਕਾਰ, ਆਗਿਆਕਾਰ
14.	ਈਆਂ	ਉੱਚਾਈਆਂ, ਬੁਰਾਈਆਂ
15.	ਿਆਂ	ਪੁਰਾਣਿਆਂ, ਪਿਆਰਿਆਂ
16.	ੀਆਂ	ਝੂਠੀਆਂ, ਠੰਡੀਆਂ

Table 4: Suffix list-4 having suffix length-2

Sr.	Suffix	Examples
1.	ੀਏ	ਸੋਹਣੀਏ, ਮੋਟੀਏ
2.	ਿਓ	ਬੁੱਢਿਓ, ਸੋਹਣਿਓ
3.	ੀਓ	ਸੋਹਣੀਓ, ਚਿੱਟੀਓ
4.	ਤਾ	ਨੈਤਿਕਤਾ, ਪਵਿੱਤਰਤਾ
5.	ਪਣ	ਸੰਘਣਾਪਣ, ਖੋਖਲਾਪਣ
6.	ਾਂ	ਸ਼ੇਕੀਨਾਂ, ਸਥਾਨਾਂ

Table 5: Suffix list-5 having suffix length-1

Sr.	Suffix	Examples
1.	ੀ	ਉਦਾਸੀ, ਉੱਘੀ
2.	ਾ	ਉੱਚਾ, ਅਭਾਗਾ
3.	ੇ	ਉੱਘੇ, ਉੱਚੇ
4.	ੂ	ਈਰਖਾਲੂ, ਦਿਆਲੂ
5.	ੋ	ਸ਼ੋਕੀਨੋ, ਸਮਝਦਾਰੋ
6.	ਈ	ਉੱਚਾਈ, ਸੱਚਾਈ

**D. Proposed Rules for PAS**

After generating the suffix lists, rules for stemming of adjectives based on the length of suffixes were developed. A total of 39 rules were created to remove the suffixes from input word.

- Rule 1: If a Punjabi adjective word ends with ‘ਪ੍ਰਸਤੀ’, strip the suffix ‘ਪ੍ਰਸਤੀ’ at the last of the input word.
- Rule 2: If a Punjabi adjective word ends with ‘ਸ਼ੀਲਤਾ’, strip the suffix ‘ਸ਼ੀਲਤਾ’ at the last of the input word.
- Rule 3: If a Punjabi adjective word ends with ‘ਪੂਰਵਕ’, strip the suffix ‘ਪੂਰਵਕ’ at the last of the input word.
- Rule 4: If a Punjabi adjective word ends with ‘ਪਾਤਰ’, strip the suffix ‘ਪਾਤਰ’ at the last of the input word.
- Rule 5: If a Punjabi adjective word ends with ‘ਪੂਰਣ’, strip the suffix ‘ਪੂਰਣ’ at the last of the input word.
- Rule 6: If a Punjabi adjective word ends with ‘ਪੁਣਾ’, strip the suffix ‘ਪੁਣਾ’ at the last of the input word.
- Rule 7: If a Punjabi adjective word ends with ‘ਸ਼ਕਤੀ’, strip the suffix ‘ਸ਼ਕਤੀ’ at the last of the input word.
- Rule 8: If a Punjabi adjective word ends with ‘ਦਾਇਕ’, strip the suffix ‘ਦਾਇਕ’ at the last of the input word.
- Rule 9: If a Punjabi adjective word ends with ‘ਬਾਜ਼ੀ’, strip the suffix ‘ਬਾਜ਼ੀ’ at the last of the input word.
- Rule 10: If a Punjabi adjective word ends with ‘ਮੰਦੀ’, strip the suffix ‘ਮੰਦੀ’ at the last of the input word.
- Rule 11: If a Punjabi adjective word ends with ‘ਬਾਜ਼’, strip the suffix ‘ਬਾਜ਼’ at the last of the input word.
- Rule 12: If a Punjabi adjective word ends with ‘ਘਾਤ’, strip the suffix ‘ਘਾਤ’ at the last of the input word.
- Rule 13: If a Punjabi adjective word ends with ‘ਯੋਗ’, strip the suffix ‘ਯੋਗ’ at the last of the input word.
- Rule 14: If a Punjabi adjective word ends with ‘ਸ਼ੀਲ’, strip the suffix ‘ਸ਼ੀਲ’ at the last of the input word.
- Rule 15: If a Punjabi adjective word ends with ‘ਵਾਨ’, strip the suffix ‘ਵਾਨ’ at the last of the input word.
- Rule 16: If a Punjabi adjective word ends with ‘ਕਰਣ’, strip the suffix ‘ਕਰਣ’ at the last of the input word.
- Rule 17: If a Punjabi adjective word ends with ‘ਉਣਾ’, strip the suffix ‘ਉਣਾ’ at the last of the input word.
- Rule 18: If a Punjabi adjective word ends with ‘ਦਾਰ’, strip the suffix ‘ਦਾਰ’ at the last of the input word.
- Rule 19: If a Punjabi adjective word ends with ‘ਹੀਣ’, strip the suffix ‘ਹੀਣ’ at the last of the input word.

- Rule 20: If a Punjabi adjective word ends with 'ਪੋਸ਼', strip the suffix 'ਪੋਸ਼' at the last of the input word.
- Rule 21: If a Punjabi adjective word ends with 'ਮੰਦ', strip the suffix 'ਮੰਦ' at the last of the input word.
- Rule 22: If a Punjabi adjective word ends with 'ਖੋਰ', strip the suffix 'ਖੋਰ' at the last of the input word.
- Rule 23: If a Punjabi adjective word ends with 'ਕਾਰ', strip the suffix 'ਕਾਰ' at the last of the input word.
- Rule 24: If a Punjabi adjective word ends with 'ਈਆਂ', strip the suffix 'ਈਆਂ' at the last of the input word.
- Rule 25: If a Punjabi adjective word ends with 'ਿਆਂ', strip the suffix 'ਿਆਂ' at the last of the input word.
- Rule 26: If a Punjabi adjective word ends with 'ੀਆਂ', strip the suffix 'ੀਆਂ' at the last of the input word.
- Rule 27: If a Punjabi adjective word ends with 'ਬੱਧ', strip the suffix 'ਬੱਧ' at the last of the input word.
- Rule 28: If a Punjabi adjective word ends with 'ੀਏ', strip the suffix 'ੀਏ' at the last of the input word.
- Rule 29: If a Punjabi adjective word ends with 'ਿਓ', strip the suffix 'ਿਓ' at the last of the input word.
- Rule 30: If a Punjabi adjective word ends with 'ੀਓ', strip the suffix 'ੀਓ' at the last of the input word.
- Rule 31: If a Punjabi adjective word ends with 'ਪਣ', strip the suffix 'ਪਣ' at the last of the input word.
- Rule 32: If a Punjabi adjective word ends with 'ਾਂ', strip the suffix 'ਾਂ' at the last of the input word.
- Rule 33: If a Punjabi adjective word ends with 'ਈ', strip the suffix 'ਈ' at the last of the input word.
- Rule 34: If a Punjabi adjective word ends with 'ਤਾ', strip the suffix 'ਤਾ' at the last of the input word.
- Rule 35: If a Punjabi adjective word ends with 'ੀ', strip the suffix 'ੀ' at the last of the input word.
- Rule 36: If a Punjabi adjective word ends with 'ਾ', strip the suffix 'ਾ' at the last of the input word.
- Rule 37: If a Punjabi adjective word ends with 'ਠੇ', strip the suffix 'ਠੇ' at the last of the input word.
- Rule 38: If a Punjabi adjective word ends with 'ਠੁ', strip the suffix 'ਠੁ' at the last of the input word.
- Rule 39: If a Punjabi adjective word ends with 'ਠੇ', strip the suffix 'ਠੇ' at the last of the input word.

### E. Proposed Algorithm

An inflected Punjabi adjective word is entered as an input to PAS (step 1). First, PAS will compare the input word with the words in the root database (step 2). If a match is found, it means that the input word is itself a root word and there is no need of stemming it. So, the same input word is provided at the output. If the input word doesn't find a match in the root database, then PAS has to go through the suffix stripping phase in order to stem the input word to its root (step 3 to step 7). Here, PAS will check all of its 5 suffix lists one by one starting with suffix list-1 which contains the longest suffixes (of length 5 characters), then suffix list-2 (of length 4 characters) and so on until it finds a matching suffix. If the ending of entered word is matched with any of suffixes from these suffix lists, then PAS removes the respective suffix from the end.

There are some special cases related to suffix list-3 (step 5), suffix list-4 (step 6), and suffix list-5 (step 7). For example, at step 5, if the input word is 'ਸੋਹਣਿਆਂ', its suffix will match with 'ਿਆਂ'. After removing the suffix, the word we will get is 'ਸੋਹਣ' which is not a valid adjective. Thus, we add 'ਾ' to the end to make it a valid adjective 'ਸੋਹਣਾ'. Similarly, at step 6, if the input word is 'ਸੋਹਣੀਓ', its suffix will match with 'ੀਓ'. After removing the suffix, the word we will get is 'ਸੋਹਣ' which is not a valid adjective. Thus, we add 'ਾ' to the end to make it a valid adjective 'ਸੋਹਣਾ'. In a similar manner, at step 7, if the input word is 'ਸੋਹਣੇ', its suffix will match with 'ੇ'. After removing the suffix, the word we will get is 'ਸੋਹਣ' which is again not a valid adjective. Thus, we add 'ਾ' to the end to make it a valid adjective 'ਸੋਹਣਾ'.

At step 8, if the input word's suffix doesn't match with any of the suffixes in the rule base, then it means that the entered word is a root word. It is therefore added to the root database and returned as output.

- Step 1. The inflected Punjabi adjective word is given as input to PAS.
- Step 2. If the entered input adjective word matches with a root word in the root database, then PAS returns the same input word as output and goes to step 9. Otherwise, go to step 3.
- Step 3. It will check suffix list-1 (length-5). If the ending of entered word is matched with any of the suffixes from the suffix list-1, then it removes the respective suffix from the end and searches the stemmed word in the root database. If the word is found then it is returned as output and PAS goes to step 9. Otherwise, go to step 4.
- Step 4. It will check suffix list-2 (length-4). If the ending of entered word is matched with any of the suffixes from the suffix list-2, then it removes the respective suffix and then searches the stemmed word in the root database. If the word is found then it is returned as output and PAS goes to step 9. Otherwise, go to step 5.
- Step 5. It will check suffix list-3 (length-3). If the ending of entered word is matched with any of the suffixes from the suffix list-3, then it removes the respective suffix. If the matched suffix of the entered word is 'ਿਆਂ' then after removing the suffix, we add 'ਾ' at the end and then search the stemmed word in the root database. If the word is found then it is returned as output and PAS goes to step 9. Otherwise, go to step 6.
- Step 6. It will check the suffix list-4 (length-2). If ending of entered word matches with any of the suffixes from the suffix list-4, then it removes the respective suffix. If the matched suffix of entered word is 'ੀਓ', 'ਿਓ', 'ੀਏ' then after removing the suffix, we add 'ਾ' at the end and then search the stemmed word in the root database. If the word is found then it is returned as output and PAS goes to step 9. Otherwise, go to step 7.
- Step 7. It will check the suffix list-5 (length-1). If the ending of entered word is matched with any of the suffixes from the suffix list-5, then it removes the respective suffix. If matched suffix of the entered word is 'ੇ',

‘◌’ then after removing the suffix, we add ‘◌’ at the end and then search the stemmed word in the root database. If the word is found then it is returned as output and PAS goes to step 9. Otherwise, go to step 8.

Step 8. If the input word doesn't match with any of the words of the root database and also its suffix doesn't match with any of the suffixes in the rule base, then it means the entered word is a root word. It is added to the root database and also returned as the output and PAS goes to step 9.

Step 9. Exit.

#### IV. RESULTS AND DISCUSSION

Accuracy of a stemmer is dependent on the size of the root database and the completeness of the rules created to remove suffixes. Bigger database results in higher accuracy [22]. We have entered 1,762 root words in our database. To analyze the performance of PAS, entered 1,762 words as input, out of which 1,564 words were correctly stemmed. Therefore, the accuracy of our stemmer is 88.76%.

#### V. CONCLUSION

The stemming system generates linguistically normalized text, which improves the result of many information retrieval applications [3]. Our stemmer works for Punjabi adjective words and this is a simplified version of a stemmer. Our system uses a rule-based suffix-stripping algorithm to stem the Punjabi adjectives. This stemmer can work as a basic pre-processing tool for Punjabi language and will be helpful in various NLP applications and in text mining.

#### VI. REFERENCES

- [1] A. S. Nayak, A. P. Kanive, Ananthu, N. Chandavekar, Naveen and B. Ramasamy, Survey on pre-processing techniques for text mining, *IJECS*6 (2016) 2319-7242.
- [2] A. Sharma, R. Kumar and V. Mansotra, Proposed stemming algorithm for Hindi information retrieval, *IJIRCCCE*4 (2016).
- [3] D. Kumar and P. Rana, Stemming of Punjabi words by using brute force technique, *International Journal of Engineering Science and Technology*3 (2011) 1351-1358.
- [4] N. Saharia, U. Sharma and J. Kalita, Analysis and evaluation of stemming algorithms: a case study with Assamese, in *Proc. ICACCI'12*, 2012, pp. 3-5.
- [5] B. Dalwadi and S. Suthar, Overview of stemming algorithms for Indian and non-Indian languages, *International Journal of Computer Science and Information Technologies*5 (2014) 1144-1146.
- [6] A. Jivani, A comparative study of stemming algorithms, *Int. J. Comp. Tech. Appl.* 2 (2011) 1930-1938.
- [7] P. Pandey, D. Amin and S. Govilkar, Rule based stemmer of Marathi wordnet for Marathi language, *IJARCCCE*5 (2016).
- [8] I. Slawik, N. Jan and A. Waibel, Stripping adjectives: integration techniques for selective stemming in smt systems, *European Association for Machine Translation* 2015.
- [9] V. Gupta, Automatic stemming of words for Punjabi language, *Advances in Intelligent Systems and Computing*264 (2014) 73-84.
- [10] R. Puri, R. P. S. Bedi and V. Goyal, Punjabi stemmer using Punjabi wordnet database, *Indian Journal of Science and Technology*, 8(2015) 1-5.
- [11] A. Ramanathan and D. D. Rao, A lightweight stemmer for Hindi, in *Proc. Workshop on Computational Linguistics for South-Asian Languages*, 2003.
- [12] H. Singh, A study of research papers on Punjabi stemming with special reference to brute force approach, *International Journal of Computer Sciences and Engineering*7 (2019) 164-167.
- [13] D. Kumar and P. Rana, Design and development of a stemmer for Punjabi, *International Journal of Computer Applications* 11 (2010) 18 – 23.
- [14] C. Dhawan, J. Singh and K. Garg, Hybrid approach for stemming in Punjabi, *International Journal of Computer Science & Communication Networks*3101-104.
- [15] V. Gupta, N. Joshi and I. Mathur, Rule based stemmer in Urdu, in *Proc. International Conference on Computer and Communication Technology*, 2013.
- [16] S. Paul, M. Tandon, N. Joshi and I. Mathur, Design of a rule based Hindi lemmatizer, *Computer Science & Information Technology*, 3 (2013) 67-74.
- [17] V. Gupta and G. S. Lehal, A survey of common stemming techniques and existing stemmers for Indian languages, *Journal of Emerging Technologies in Web Intelligence*5 (2013).
- [18] S. Dasgupta and V. Ng, Unsupervised Morphological Parsing of Bengali, *Language Resources and Evaluation*, 40(2006) 311-330.
- [19] J. Goldsmith, Unsupervised learning for morphology of natural languages, *Computational Linguistics* 27(2001) 153-198.
- [20] J. Singh and V. Gupta, A systematic review of text stemming techniques, 48 (2016) 157–217.
- [21] P. K. Patel, P. Kashyap and P. Bhattacharyya, Hybrid stemmer for Gujarati, in *Proc. 23rd International Conference on Computational Linguistics (COLING)*, 2010.