# Mining Text for Meaningful Words with Stemming Algorithm

Priti Shende
Department of Computer Science and Engineering
Rashtrasant Tukadoji Maharaj Nagpur University
Nagpur, India

V. B. Kute
Department of Computer Engineering
Rashtrasant Tukadoji Maharaj Nagpur University
Nagpur, India

## ABSTRACT
With the growth of explosive Internet information, data availability is easy. However, raw data is useful when mined. Therefore, mining is an important research area. The text mining primarily aims at discovery and retrieval of useful and interesting patterns from a large database. Identification and understanding of appropriate words is important to retrieve appropriate documents. Referring dictionary is time consuming and tedious job for understanding meaning of words every time. This can be prevented by converting different occurrences of word forms to its root. Frequency of words occurrences in a file used to prioritized documents. This works target avoidance of incomplete and meaningless words generation using stemming. We propose a method to compare different forms of words present in the document up to certain length. Sixty percent length of the word considered for comparison. Words having common letters are considered as different forms of same root.

## General Terms
Data Mining, Text Mining, Porter Stemmer

## Keywords
Complete words, sixty percent length, Porter's stemming algoithm

## 1. INTRODUCTION
Stemming is the preprocessing method used to reduce the words into their root form. Reducing words is nothing but removing suffixes that leads to make all the different forms of word into the same root. Stemming gives not only incomplete words but also under-stemming and over-stemming. Converting various forms of same word into different roots is called as Under-stemming. Over-stemming means converting form of different words into same root. Stemming algorithms are classified into three groups namely Truncating method, Statistical method and Mixed method. Many algorithms are emerged and belong to one of the methods. But Porter's algorithm is most suitable for stemming due to its low error rate. It proved best as compare to other algorithms due to its output [1]. Many improvements have been done on basic Porter's algorithm [2]. But increment of rules in it, increases under-stemming. Large database of rules consumes time. Clustering algorithms can be used to find the stem among the members in a group. The stem is one whose position is at the center of the group. Choosing characteristics of words for group formation and threshold value is very difficult task [3].In the proposed method, words present in the document are preferred for stemming purpose. Initially few characters are matched and condition satisfied words are considered in one group. Words in a group are again compared up to new-length (N) i.e. 60% of length of the word having greatest length. Those words that are having common letters up to 60% length are converted into same form. The word chosen for converting all the forms is the one having maximum frequency in the group. Words that are included in the group but don't have common letters up to N again form new group. Now N will be calculated according to the words in new group. Words are compared and those words that do not satisfied the rule again forms new group. The formation of groups continues until either all the words satisfy the condition after forming group(s) or a single word is remained. At last, words converted into one of the forms.

## 2. PORTER'S STEMMING ALGORITHM
Porter's stemming algorithm is a base of stemming algorithm. It is a preprocessing technique used for text mining. Number of algorithms is developed based on it. Many algorithms compared the efficiency of their algorithm with Porter stemmer [4]. Efficiency can be measured in terms of under-stemming, over-stemming, precision, recall and also word completeness. Porter stemmer is widely referred and also implemented in other languages [5]. Porter stemmer removes suffixes from the word. It is a rule based approach and has 5 steps consists of 60 rules. The algorithm mainly focused on conditions such as vowel-consonant pair or at least one vowel or double consonant. Initially suffixes are checked and conditions are applied if a word follows given suffixes in the rule. Now the word is partially or fully stemmed according to the suffix attached to it. The same word is forwarded to next step for applying rules on it. If a word follows the rules of that step then suffix is removed again. So, this process continues until step 5. At the end stemmed words are available [6].Sometimes for words like "knowingness", "ness" is removed but "ing" is not removed. Because rule to remove "ing" is in step 1 and that of "ness" is in step 3. Word has "ness" at the end till that time rule for removing "ing" is gone. At last the stemmed word is "knowing". For the original word "knowing" occurred in the document, "ing" is removed and word stemmed as "know". This is under-stemming as two forms of same word are having different stems i.e. "knowing" for word "knowingness" and "know" for word "knowing".

## 3. PROCEDURE OF TEXT EXTRACTION
Articles of International Journal of Computer Applications are taken in PDF form as input. These articles are converted into HTML format by using readymade tool. Text is extracted in WORD document from "ABSTRACT" section till "REFERENCES" section of articles for analysis. Special characters and common words such as "is", "their", "after" are removed [7]. Sorting algorithm is applied for the immediate occurrences of different forms of same word. Proposed algorithm is applied for stemming. Such steps are taken for stemming the words in the document. This can help us while comparison of two documents according to query fired. As proposed algorithm works on the words present in the document, it is not necessary that all the forms present in

document. Some forms of word can present in one document and another form in another document. In such case word present in the query is compared with the forms of word available in the document. Words have some common characters up to certain length. Starting four characters are mostly common [8]. So, proposed algorithm has utilized the advantage of words having some common characters. Rule 2nd is applicable if a word in query ends with proper suffix and word of 3 or 4 letters are formed after subtracting suffix from the original word. Otherwise initial 5 characters are matched with the words present in the document. If matched then rule 5 is applicable.

# 4. PROPOSED METHOD

Proposed method is a method in which stemming of words is done on the basis of statistics. One formula is designed on the basis of length of word. That word will be the one having maximum length in a group. New-length (N) for comparison of words is calculated using the formula "(1)".

$$N = 0.6 * L, \quad (1)$$

where L is the length of word.

Algorithm is divided into steps according to the length of words. Following are the steps.

## 4.1 STEP 1

Words less than three characters are remaining as it is in the document. Conversion of words should not be done.

## 4.2 STEP 2

### *4.2.1 CASE 1: Root form is available with other forms of word in a file*

Words having length 3 or 4 characters are recognized. They are compared with the words having all the characters common and end with proper suffix included in the list. If characters are same up to 3rd or 4th position then group is formed. List of Suffixes is

"s", "ing", "ed", "ness", "ly", "by", "ion", "ize", "ant", "ent", "ic", "al", "Ic", "ical", "able", "ance", "ary", "ate", "ce", "y", "dom", "ee", "eer", "ence", "ency", "ery", "ess", "ful", "hood", "ible", "icity", "ify", "ish", "ism", "ist", "istic", "ity", "ive", "less", "let", "like", "ment", "ory", "ous", "ty", "ship", "some", "ure", "n" [1].

IF words have 3 or 4 letters common after subtracting length of suffix attached to them from their length then the words belong to same root. Word "ebb" is having 3 letters. So all the characters are matched with "ebbing" and "ebbed" and they have proper suffix "ing" and "ed". While subtracting the length of suffix from the words "ebbed" and "ebbing", word of 3 letters "ebb" is formed. So, all the three words belong to same root. All the words are replaced by one word having maximum frequency. Maximum frequency is of word "ebbing" as shown in Table 1.

**Table 1. Words belong to Case 1 of Step 2**

| Word | Length | Frequency |
|---|---|---|
| ebb | 3 | 12 |
| ebbed | 5 | 10 |
| ebbing | 6 | 14 |

Result after applying rule is shown in Table 2.

**Table 2. Words after the application of rule**

| Word | Frequency |
|---|---|
| ebbing | 12 |
| ebbing | 10 |
| ebbing | 14 |

At the end, all frequencies are added and total frequency is equal to 36 as shown in Table 3.

**Table 3. Final stemmed word**

| Word | Frequency |
|---|---|
| ebbing | 36 |

If words of 3 or 4 characters are not present, but their other forms are present then recognition of such words is also necessary. If a word having length greater than 2, less than 10 and ends with proper suffix then subtract the suffix from the word. If 3 or 4 characters are found then compare its initial 3 or 4 characters with the other words end with proper suffix. Form a group and replace words with a word having maximum frequency if they have common initial characters.

### *4.2.2 Case 2: Different forms of word except root word are available in a file.*

Consider an example of words "knowing" and "known". Words are having proper length and suffix. Subtracting the length of suffixes "ing" and "n" from the respective words and comparing the words gives 4 characters common. Maximum frequency is 20 as shown in Table 4.

Replacement is done by word "known" and total frequency is 37 as shown in Table 5.

**Table 4. Words belong to Case 2 of Step 2**

| Word | Length | Frequency |
|---|---|---|
| knowing | 7 | 17 |
| known | 5 | 20 |

**Table 5. Final stemmed word**

| Word | Frequency |
|---|---|
| known | 37 |

### *4.2.3 Case 3: Different forms of same word are available in different files.*

Suppose 1st file contains word "adding" has frequency 10 and 2nd file contains word "add" has frequency 15. The query word is "added". Comparison of word "added" is done with words "adding" and "add". All the three words are having length below 9, initial 3 characters are common and end with proper suffix. 3 letters are obtained after subtracting the length of suffix from the length of word. So, all the words belong to same form. 2nd file is having higher priority due to its maximum frequency i.e. 15 as shown in Table 6.

**Table 6. Words belong to Case 3 of Step 2**

| Word | Length | Frequency |
|---|---|---|
| adding | 6 | 10 |
| add | 3 | 15 |
| added | 5 | |

Words that don't follow the above rule and length greater than 5 fall under the STEP 3.

## 4.3 STEP 3

### *4.3.1 Case 1: Forms of word present in a file*

Words greater than or equal to 5 characters are recognized. Initial 5 characters are matched and group is formed. Word having largest length is found and N is calculated using Equation. 1. Characters are again matched up to N. Words that have common letters are replaced with the word having maximum frequency among them. Words that don't have common letters up to N are again form new group. N is again calculated according to word having greatest length in the new group. This process continues until each and every word satisfies the condition.

5 characters are matched and L of word "arriving" is largest i.e. 8 as shown in Table 7.

$$N = 0.6 * 8 = 4.8 = 4 \quad (2)$$

According to Equation 2, N is equal to 4, mantissa is not taken. Up to 4th position i.e. letter "i" all the characters are same. So, these words belong to one stem. All the words are replaced by "arrival" as shown in Table 8.

**Table 7. Words belong to Case 1 of Step 3**

| Word | Length | Frequency |
|---|---|---|
| arrival | 7 | 20 |
| arrive | 6 | 9 |
| arriving | 8 | 12 |

**Table 8. Words after the application of rule**

| Word | Frequency |
|---|---|
| arrival | 20 |
| arrival | 9 |
| arrival | 12 |

Final result is shown in Table 9. Total frequency is equal to 41.

**Table 9. Final stemmed word**

| Word | Frequency |
|---|---|
| arrival | 41 |

### *4.3.2 Case 2: Forms of word present in different files*

Suppose 1st file contains word "network" has frequency 11 and 2nd file contains word "networks" has frequency 5 as shown in Table 10. The query word is "networking". Comparison of word "networking" is done with words "network" and "networks". Initial 5 characters are common and 6 is the calculated length as shown in Equation 3.

$$N = 10 * 0.6 = 6 \quad (3)$$

**Table 10. Words belong to Case 2 of Step 3.**

| Word | Length | Frequency |
|---|---|---|
| network | 7 | 11 |
| networks | 8 | 5 |
| networking | 10 | |

Characters up to length 6th position i.e. "r" are matched. All three words are having common characters up to "r" and considered as the different forms of same stem. As word "networks" has greater frequency than word "network". So, 1st file has higher priority than 2nd file.

## 5. EVALUATION OF RESULTS

Porter stemmer is a strong stemmer to convert the words in to their root form. But for words like "drayman" and "draymen" or "child" and "children" root words are different. This is called under-stemming. Proposed algorithm creates proper roots for such words. Example of each category under words fall is given in [2]. In such cases proposed algorithm gives proper root. The comparison of both the algorithms for different groups is shown in Figure 1. Group includes various forms of a word. Stronger stemmer produces less stems than words [9] and various forms of word assigned to one class is the measure of stronger stemmer. Index compression factor is inversely proportional to roots. Stronger stemmer should give large value of it. Proposed algorithm forms lesser roots than Porter's algorithm.

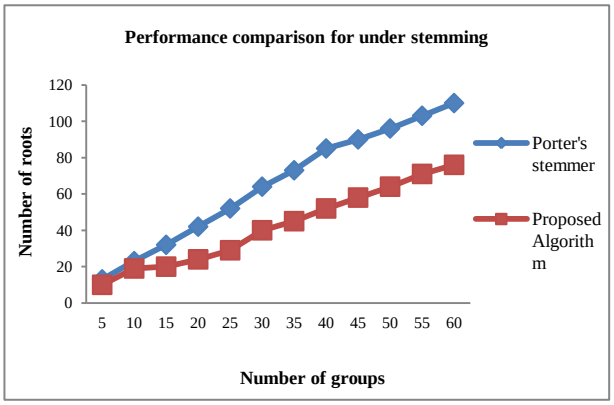


**Fig. 1: Performance comparison of under stemming**

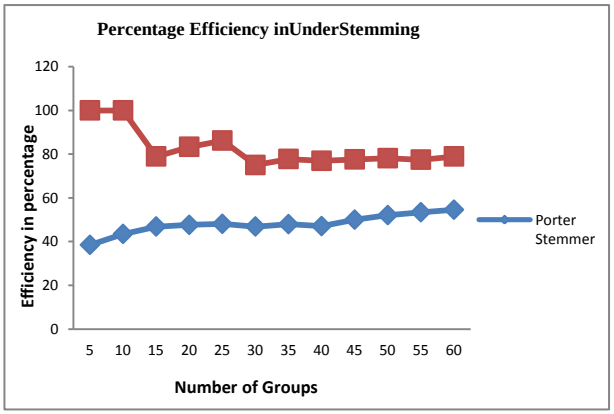


**Fig. 2: Percentage efficiency in under-stemming**

Instead of taking complete groups, comparison can also be done in terms of words. From each group few words are taken or a single word. In this case, proposed algorithm works better as it forms lesser roots than Porter's algorithm as shown in Figure 3. In single form of word proposed algorithm always performs better and gives complete word.

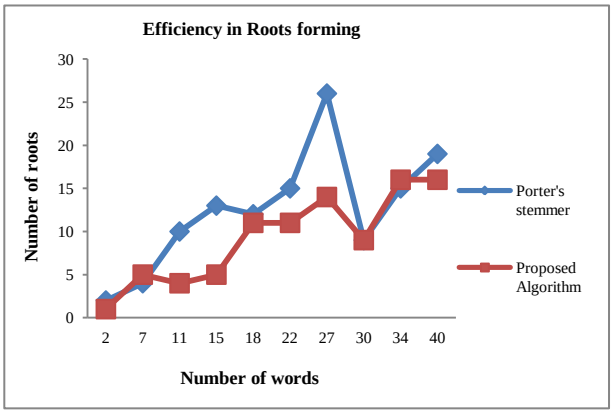


**Fig. 3: Comparison of roots forming**

If a single word of one class is present in the document then that word is not replaced with any word and that word will remain as it is.

## 6. CONCLUSION

Though Porter stemmer is a powerful algorithm for creating roots but often gives incomplete words. It has 60 rules for stemming, each word is gone through 5 steps and requires large database. Proposed algorithm has no such set of rules and hence only requires database for files storage. It gives 100% meaningful words and time required is nearly same or less than Porter's algorithm. Under stemming is less for words having length more than 6. If different forms of a word are not present then word has no change in it. So, proposed algorithm performs better than Porter's algorithm.

## 7. FUTURE SCOPE

Proposed algorithm compares words by having common characters in it. Sometime different words have 80% common letters. Words like "GENERATE" and "GENERAL" has common characters up to "A". Hence, they stemmed into one of the words according to frequency. Improvement can be done in the existing formula. Addition of new formula or combining that with the existing algorithms can be done.

## 8. REFRENCES

[1] Ms. Anjali Ganesh Jivani, "A comparative study of Stemming algorithms", in Int. J. Comp. Tech. Appl., Vol 2 (6), 1930-1938

[2] Wahiba Ben Abdessalem Karaa, "A new stemmer to improve information retrieval", in International Journal of Network Security And Its Applications(IJNSA), Vol. 5, No. 4, July 2013

[3] Prasenjit Majumder, Mandar Mitra, Swapnil K. Parui and Gobinda Kole , Pabitra Mitra and Kalyankumar Datta, "YASS: Yet Another Suffix Stripper", ACM transactions on information systems, vol. 25, no. 4, article 18, publication date: October 2007

[4] K.K. Agbele, A.O. Adesina, N.A. Azeez , & A.P. Abidoye, "Context-Aware Stemming algorithm for semantically related root words", in African Journal of Computing & ICT Vol 5. No. 4, June 2012

[5] Peter Willet, "The Porter stemming algorithm: then and now", in electronic library and information systems, 40(3).pp. 219-223

[6] M. F. Porter, "An algorithm for suffix stripping", Originally published in Program, Vo1. 4 no. 3, pp 130-137, July 1980.

[7] Danilo Saft and Volker Nissen, "Analysing full text content by means of flexible co-citation analysis inspired text mining method- exploring 15 years of JASSS articles", Int. J. Business Intelligence and Data Mining, Vol. 9, No. 1, 2014

[8] B. P. Pande, Pawan Tamta, H. S. Dhami, "Generation, Implementation and Appraisal of an N-gram based Stemming Algorithm", in press

[9] William B. Frakes, Christopher J. Fox, "Strength and similarity of affix removal stemming algorithm", in press