ISSN: 2348 - 2117

International Journal of Engineering Technology and Computer Research (IJETCR)
Available Online at www.ijetcr.org
Volume 2 Issue 5 Page No. 40-44

A study of stemming algorithms in text mining

Dr. (Ms). Ananthi Sheshasayee[1], Ms.G. Thailambal[2]

[1]Head and Associate Professor, Quaid -e- Milleth College for Women, Chennai, India
ananthi.research@gmail.com

[2]Research Scholar, SCSVMV University, Kancheepuram, India
thaila.research@gmail.com

ABSTRACT
Text Mining is the term used as an alternate to Knowledge Discovery in Web. Finding relevant documents for the user query is the process of text mining. Deciding which documents should be retrieved is a difficult task for the researcher. The query terms are matched along with the terms of the document and the decision is made either binary, i.e., Retrieved / Rejected or the relevance of the document is estimated. The terms will be morphological so that the need of stemming arises which further reduces the size of the document. Therefore, it increases the processing time and decreases the storage space. In this paper a study of Stemming algorithms and the factors used to calculate its efficiency and the strength of the Stemmer are discussed.

Key words: *Text Mining, Stemming, Morphological terms, Porter Stemmer*

INTRODUCTION:

Text mining is used to describe the application of data mining techniques to automated discovery of useful or interesting knowledge from unstructured text. Stemming is applied in the pre-processing task of text mining applications. Several separate algorithms have been framed for many Languages. [1]. Pre-processing is important since it is necessary to obtain all the words given in a query text document split into tokens called 'Tokenization'. In tokenization some standard methods are used such as Separation on white spaces, Alphabetic Strings and Alphanumeric Strings [2]. Merging of different words taken from different documents placed in 'Dictionary' for future verification. To reduce the size of the dictionary methods such as Stemming, Stop Word Removal are used which increases the efficiency of searching the dictionary.

Word Stemming is used to reduce the number of words in the documents which will in turn reduce the processing time. Most of the time Morphological terms of words have the same semantic meaning with similar interpretation, i.e. the meaning is same and the word form is different it can be changed to its base form which is an important feature in text mining applications. Stemming Algorithms are mainly used to reduce the size of the file and to improve the efficiency of searching. In Information Retrieval process 50% of file size could be reduced if stemming is applied instead of using terms. There are two ways of identifying errors in stemming process namely Overstemming or Under stemming [12]. When Over stemming occurs much of the terms are stemmed and due to that Non-Relevant documents are extracted. When under stemming occurs only very few terms are stemmed and hence relevant documents are not extracted.

Figure 1 describes the steps of pre-processing which are extraction, stemming and Stop word Removal. After applying these pre-processing techniques the document is sent for further process of Mining [3].

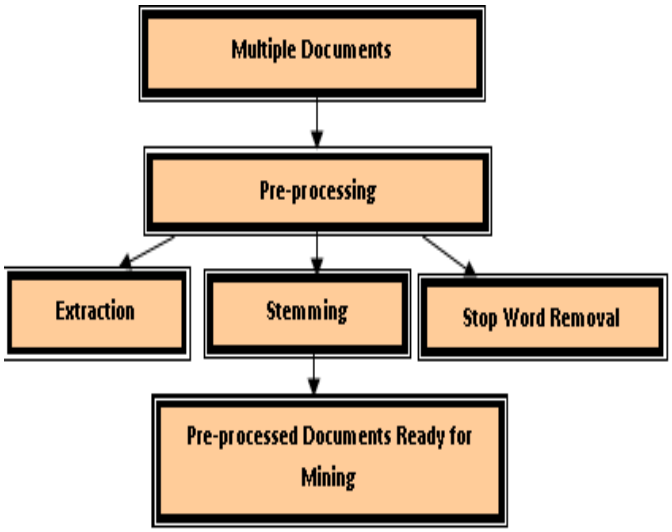

Figure 1: Pre-Processing Steps

Table 2 describes the root words of Corresponding terms. For all the term **'Running, Run, Ran'** the meaning is same and the root word is also same as **'Run'**.

Table 1: Term and its Stem Words

Term	Stem
Running	Run
Ran	Run
Run	Run

1. TAXONOMY OF STEMMING ALGORITHMS:

Different Varieties of algorithms available in stemming classified into Brute Force, Suffix Stripping and Lemmatisation [4]. Brute Force algorithms contain a lookup table which compares with the word and identifies the root. A smaller list of rules is provided and the terms are stemmed, according to that rule. Lemmatisation is identifying the correct lexical category of words, applying different Normalization rules are followed by stemming rules. The only difference between stemming and lemmatisation is that the former will not consider POS (Parts of Speech). There are two important factors that need to be considered while stemming which is done only when the morphological terms have the same base meaning and the words which do not have the same meaning are maintained. Based on the process of identifying stem from the word stemming is divided into three categories Truncating methods, Statistical Methods and Mixed methods.[14]

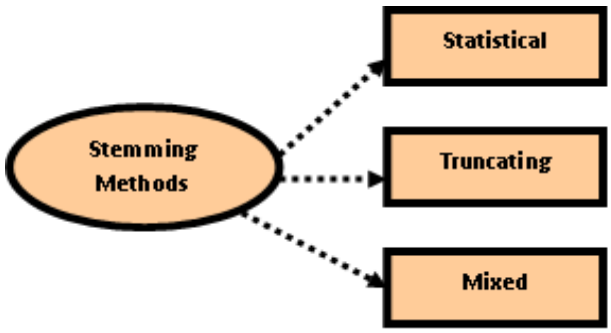

Figure 2: Methods of Stemming

Each method uses separate algorithms. In this paper, it is concentrated only on Truncating and Statistical methods. In truncating some of the suffixes or prefixes are removed. In Statistical methods suffixes are removed after applying some statistical methods.

Four [5] approaches used in the above types of methods are:

A) Affix Removal

B) Successor Variety

C) Table Look Up

D) N-grams

A. Affix Removal: This Method removes Suffixes or Prefixes from the term so that only the stem is left. An example for this type of a stemmer is removing plurals from terms. A set of rules for this type framed by Harman [13] 1991 as

(i) If a word ends in 'ies' not 'eies' or 'aies' it should be replaced with 'y'

i.e., Cries -> Cry, Berries-> Berry

(ii) If a word ends in 'es' not aes'or'ees'or 'oes replaced with 'e'.

(iii) If a word ends in 's' not in 'us' or 'ss' replaced with 'null'.

i.e., Bicycles-> Bicycle, Clones-> Clone

The Second type of Stemmer algorithm is Porter Stemming Algorithm [11]. This algorithm contains three classes of conditions (i) Conditions on the Stem (ii) Conditions on the Suffix (iii) Conditions on the rules.

(i) Types of stem conditions as follows

Type 1: Let C denote Consonants and V

Denotes Vowels

$[C](VC)^m[V]$

Superscript 'm' indicates number of 'VC' Sequences

m=0 i.e. 'EE'

m=1 i.e. 'TREES'

m=2 i.e. 'PRIVATE'

Type 2: *<X> - The stem ends with the letter 'X'

Type 3:*v* - The stem contains a Vowel

Type 4:*d – The stem ends in Double consonant

Type 5:*0- The stem ends with the Consonant-Vowel-Consonant sequence and not 'w','x','y' in last consonant.

(ii) Suffix Conditions takes the form of (Current Suffix==>Pattern)[6]

The five steps of porter algorithm are as follows:

Step 1: Reduces the Plurals and –'ed' or –'ing' Suffixes

Step 2: When another Vowel presence in the stem then it changes 'y' in terminal to 'i'.

Step 3: Maps Double suffixes to single ones such as –'ization' , -'ational'.

Step 4: Changes suffixes –'full', -'ness'.

Step 5: Removes –'ant', -'ence'.

Step 6: Removes final –'e'.

It has some disadvantages which are improved by applying some additional rules to the previous algorithm. The changes are

(i) Terminating 'y' changed to 'i'.

(ii) Suffix 'us' does not lose its 's'.

(iii) Removal of additional suffixes, including 'ly'.

The errors are minimized by applying these changes.

B. Successor Variety:- It uses the sequence of terms appearing in the text to apply stemming. This method is based on Morphemes and Phonemes in large body of text utterances where Morpheme is unit of a language that cannot be subdivided and Phonemes are sounds through which one word is distinguished from others.

The steps followed in finding Successor Variety according to Hafer and Weiss [10] is:

Step 1: Let α be a word of length n.

Step 2: αi is a length i prefix of α.

Step 3: Let D be the corpus of words.

Step 4: D_i is defined as the subset of D Containing those terms whose first i letters match i exactly.

Step 5: Number of distinct letters that occupy the $(i + 1)^{th}$ position of words in D_i.

A test word of length n has n successor varieties $S_\alpha i$, $S_\alpha 2$, . . . , $S_\alpha n$. In turn the Successor variety of String is the number of characters that follow it in words in the text. For example Corpus of words used Running, Ran, Run, Rose, Ride, Rise, Runs. Consider a Test word **RUIN**. The successor term is given as

Prefix αi	$D_\alpha i$	Successor Variety $S_\alpha i$
R	Running, Ran Run, Rose, Ride, Rise, Runs	4(U,A,O,I)
RU	Running, Run Runs	1(N)
RUI	Empty	0

Table 2: Successor Varieties of the word RUIN

If it is applied to a large set of text the successor variety increases since more characters will be added to the segment which is used to identify words for Stemming. After identifying the words any of the following four methods can be used.

i) Cutoff method:-Cutoff Value is selected for successor varieties and if it is reached a boundary value is selected. If it is small cutoff value proper cuts cannot be made. If it is large many cuts will be missed. Hence the correct cutoff value is selected for obtaining efficient results.

ii) Peak and Plateau Method:-A better method than cutoff method is the current method since no need to select cutoff value. Whenever a character immediately preceding and immediately following it in the successor variety exceeds a segment break is made.

iii) Complete word Method:-When the Segment chosen is a complete word in the corpus a segment break is made.

iv) Entropy Method:-A set of entropy measures defined in a word and also for a predecessor and the cutoff value is also measured along with identifying boundaries. Let $|D_\alpha i|$ be the number of words in a text with 'i' length sequence of letters. Let $|D_\alpha ij|$ be the number of words in 'Di' with the successor 'j'[15].

The Probability that a member of Di has the successor j is given by

$$\frac{|D_\alpha ij|}{|D_\alpha i|} \quad \text{--- (1)}$$

The entropy of $|D_\alpha i|$ is

$$H\alpha i = E - \frac{|D_\alpha ij|}{|D_\alpha i|} \text{ Log } 2 \frac{|D_\alpha ij|}{|D_\alpha i|} \quad \text{--- (2)}$$

The successor variety stemming process has 3 steps to follow:

(i) Determine the successor variety for a word

(ii) Use any one method and when the successor Variety found Segment the word.

(iii) Select one of the segments as Stem.

C. Table Look-up:-Terms and their corresponding stems are stored in the table and it is used for stemming using lookup in the table. Using B-tree and Hash table lookups are made faster. Many Delimits in using these types of table lookups such as

a) Some terms are not Standard English terms which could also be used does not contain Stemming method.

b) Storage of all the terms and its stem is overhead.

c) Timing for retrieval depends on the amount of storage of terms and stems.

D. N-Grams:-This method uses digrams and N-grams used in the text for stemming purpose. Digram is a pair of consecutive letters. An association between these pairs of letters are identified.

For Example the terms

Numeric -> Nu um me er ri ic

Unique Digrams -> Nu um me er ri ic

Numerical -> Nu um me er ri ic ca al

Unique Digrams -> Nu um me er ri ic ca al

Thus the word Numeric has 6 unique digrams and Numerical has 8 unique digrams. The two words share 6 Unique Digrams and from which similarity measure is identified. It is calculated using Dice's Coefficient

$$S = \frac{2*C}{A+B} \quad \text{--- (3)}$$

Where A is the number of unique digrams in the first word and B is the number of unique digrams in Second word and C is the share unique digrams.

So for the above example, it is calculated as

S= 2 * 6/6+8= 0.85

The Similarity measure is calculated for all the words and a triangular similarity matrix (**Sij=Sji**) is formed.

$Word_1$ $Word_2$ $Word_3$..... $Word_{n-1}$

$Word_1$

$Word_2$ S21

$Word_3$ S31 S32 ...

$Word_{n-1}$ Sn1 Sn2 Sn3 Sn (n-1)

After Forming this Single Link Clustering method is used to cluster the terms.

2. EVALUATION OF STEMMER ALGORITHMS: To find the Strength and the accuracy of the Stemmer the following criteria are measured. The degree to which a stemmer changes words that it stems is called Stemmer strength [8].

(i) Index Compression Factor (ICF):-Collection of distinct words reduced by stemming is calculated using **ICF**. When more number of words stemmed, strength of the stemmer is increased.

$$ICF = \frac{(N-S)}{S} * 100 \quad \text{--- (4)}$$

Where **N** is the number of distinct words before stemming and **S** is the Number of distinct stems after stemming.

(ii) Word Stemmed Factor (WSF):- Number of words stemmed out of the total number of words percentage is **WSF**.

It is calculated as

$$WSF = \frac{WS}{TW} * 100 \text{ --- (5)}$$

Where **Ws** is number of words stemmed and **TW** is total number of words in the sample.

(iii) Correctly Stemmed Words Factor (CSWF):- Percentage of Number of words Stemmed correctly is calculated using

CSWF. If the percentage is higher the accuracy is also higher. It is calculated as

$$CSWF = \frac{CSW}{WF} * 100 \quad \text{--- (6)}$$

Where **CSW** is the number of correctly stemmed words and WF is Total number of words Stemmed.

(iv) Average Words Conflation Factor (AWCF):-Average number of variant words with different conflation group is stemmed correctly to the root words which are calculated by first finding the number of words after conflation as

NWC = S – CW --- (7)

Where **CW** is correct number of words which were not stemmed. The word conflation factor is calculated as

$$AWCF = \frac{CSW - NWC}{CSW} * 100 \quad \text{--- (8)}$$

The Percentage is directly proportional to the strength of the stemmer.

3. CONCLUSION:

Finding the root using stemming algorithms is very difficult as no language follows proper rules. Two metrics have been followed to find the correct stemming algorithms Recall and Precision [9]. Recall indicates the relevant instances that are retrieved and Precision indicates retrieved instances that are relevant. Mostly short content will get higher results in applying stemming algorithms and highly inflictive languages get benefited since their morphological terms are more related. Many Evaluation Factors are also discussed which helps to find the strength of the Stemmer. The efficiency of the algorithm depends on the number of words stemmed. Different Stemmer algorithms are used for different languages in which certain stemmer algorithms have some disadvantages which can be corrected by combining some of the stemming algorithms or appending rules for their efficiency.

4. REFERENCES:

1. M. Thangarasu et.al. “A Literature Review: Stemming Algorithms for Indian Languages”, International Journal of Computer Trends and Technology (IJCTT) - volume 4 Issue 8-August 2013.
2. Noraida Haji Ali et.al. “Porter Stemming Algorithm for Semantic Checking”, International Journal of Computer Applications, vol. 5, pp. 15-. 25. 2010.
3. R. Ramya et.al., “Effective Pre-Processing Activities in Text Mining Using Improved Porter’s Stemming Algorithm”, International Journal Of Advanced Research in Computer and Communication Engineering Vol. 2, Issue 12, December 2013, ISSN (Print): 2319-5940, ISSN (Online): 2278-1021.
4. Giridhar N .S. et.al., “A Prospective Study of Stemming Algorithms for Web Text Mining**”,** Ganpat University Journal Of Engineering & Technology, Vol. -1, Issue-1, Jan-Jun-2011.
5. Deepika Sharma. “Stemming Algorithms: A Comparative Study and their Analysis”, International Journal of Applied Information Systems (IJAIS) - ISSN: 2249-0868 Foundation of Computer Science FCS,New York, USA, Volume 4- No.3, September 2012.
6. Anjali Ganesh Jivani et al,” A Comparative Study of Stemming Algorithms**,”** International Journal of Comp. Tech. Appl., Vol 2 (6), 1930-1938, 2011.

7. William B. Frakes et.al., "Information Retrieval: Data Structures and algorithms" Book "Chapter 8: Stemming Algorithms".
8. Sandeep R. Sirsat et al**,** "Strength and Accuracy Analysis of Affix Removal Stemming Algorithms", (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 4 (2), 2013, 265 - 269, ISSN:0975-9646.
9. Wahiba Ben Abdessalem Karaa, "A New Stemmer to Improve Information Retrieval", International Journal of Network Security & Its Applications (IJNSA), Vol.5, No.4, July 2013.
10. HAFER, M., and S. WEISS. 1974. "Word Segmentation by Letter Successor Varieties," Information Storage and Retrieval, Chapter 10, 371 - 85.
11. Porter, M.F. (2006)," An algorithm for suffix stripping", Program: electronic library and information systems, Vol. 40 Iss: 3, pp.211 – 218.
12. Wahiba Ben Abdessalem Karaa, "A New Stemmer To Improve Information Retrieval", International Journal of Network Security & Its Applications (IJNSA), Vol.5, No.4, July 2013.
13. Donna Harman,"How Effective Is Suffixing?", Journal Of The American Society For Information Science. 42(1):7-15, 1991.
14. S.Santhana Megala, et.al., "Improvised Stemming Algorithm – TWIG", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 7, July 2013 ISSN: 2277 128X.
15. Riyad Al-Shalabi et.al., "Experiments with the successor Variety Algorithms Using the CutOff and Entropy Method", Information Technology Journal 4(1):55-62, 2005. ISSN: 1812-5638.