Performance Enhancement of Stemming using Enhanced Context Aware Stemming Technique for Knowledge Discovery

Dr.B.Ramesh, Assistant Professor & Head, Department of Computer Science, Sudharsan College of Arts and Science, Pudhukkottai, TN, India. ram.73110@gmail.com

Abstract Stemming technique is used to reduce words length to their origin form, by removing derivational and inflectional affixes. The main aim of stemming is to find stem of particular word. There are several stemming techniques available in present such as Porter, Lovins, Paice's husk and affix word removal. However, recent stemming techniques have several drawbacks such as over-stemming and under-stemming, since its simple rule cannot fully describe English morphology. The performance of information retrieval may reduce the errors of stemming technique. In this research work, introduced Enhanced Context Aware Stemming algorithm to improve the performance of stemming. The proposed Enhanced Context Aware Stemming algorithm examined with standard datasets. The performance of Enhanced Context Aware Stemming algorithm is better than others.

***Keywords* — Pre-processing, Stemming Techniques, ECAS, Knowledge Discovery.**

I. INTRODUCTION

Traditional text classification techniques become inadequate for the increasingly vast amount of text data. A typical text mining problem is to locate relevant documents from a huge document collection. User need tools to compare different documents rank the importance and find patterns and trends across multiple documents. Hence Text mining plays a vital role in the Information retrieval systems.

The main objective of pre-processing is to obtain the key features or key terms from stored text documents and to enhance the relevancy between word and document and the relevancy between word and category. Pre-Processing step is crucial in determining the quality of the next stage, that is, the classification stage. It is important to select the significant keywords that carry the meaning and discard the words that do not contribute to distinguishing between the documents. The pre-processing phase of the study converts the original textual data in a data-mining ready structure. Stemming is a step in processing textual data preceding the tasks of information retrieval, text mining, and natural language processing. Thus, it is an important feature supported by present day indexing and searching systems.

Stemming algorithms reduce different morphological variants to their base form (the stem). Stemming is used to enable matching of queries and documents in keyword-based information retrieval systems. This assumes that morphological variants of words have similar semantic interpretations and can be considered as equivalent for the purpose of IR applications. It is for this reason, that stemming algorithms, or stemmers, have been developed, which attempt to reduce a word to its stem or root form.

Paik et. al., discussed Stemming [1] is the conflation of the variant forms of a word into a single representation, For example, the terms presentation, presenting, and presented could all be stemmed to present. The stem does not have to be a valid word, but it needs to capture the meaning of the words. Stemming is usually carried out by algorithms that strip word suffixes (but some also strip prefixes) which is why this technique is called affix stripping. Other stemming techniques include the use of dictionaries – which contain the correct form of stemming for the maximum number of words –and statistical stemming. Affix stripping stemmers are language dependent, that is, the rules are designed based on some knowledge of the language. One cannot use stemming rules de- signed for Portuguese, for example, and expect them to perform well on a French corpus. Statistical stemmers, however, aim at learning the stemming rules automatically and thus eliminating the need of knowing the language.

In tokenization, some stop words, such as "the", "a", will be removed as these words provide little useful information. Information Retrieval is essentially a matter of deciding which documents in a collection should be retrieved to satisfy user's need of information. Conflation is the process of merging or lumping together non identical words which refer to the same principal concept. Stemming is a fundamental step in processing textual data preceding the tasks of information retrieval, text mining, and Natural Language Processing. The common goal of

 DOI : 10.18231/2454-9150.2018.1341

stemming is to standardize words by reducing a word to its base. Data mining techniques are very useful to manipulating and analyzing data from database.

II. LITERATURE REVIEW

Stemming is a well-known research problem but in early days of stemming it was studied only for English language. Lovin's stemmer is one of the oldest stemmer developed for English using context sensitive longest match technique. Text mining refers generally to the process of extracting interesting information and knowledge from unstructured text. Text mining is the process of discovering information in text documents. Brajendra Singh Rajput [2] studied a variety of stemming methods and got to know that stemming appreciably increases the retrieval results for both rule dependent and statistical approach. It is also useful in reducing the size of index files and feature set or attribute as the number of words to be indexed are reduced to common forms called stems. The performance of statistical stemmers is far superior to some well-known rule-based stemmers but time consuming. Rule dependent stemmer like porter stemmer is good choice for English document processing but its language dependent. S.P.Ruba et al. [3] analyzed performance of several stemming techniques to text classification. Also they are shows limitations of existing stemming techniques. Finally, they are concluded porter stemming techniques is used to reduce the time and improved classification accuracy.

Stefano et al. [4] proposes a methodology to automatically learn linguistic resources for a natural language starting from texts written in that language. This methodology is examined with several languages such as English, Italian, French and Latin. However, need to run more experiments to evaluate the performance of high-level NLP tasks based on learned resources. Wahiba et al. [5] proposed improved version of the porter stemmer called new stemmer for rectifying the limitations of porter stemmer algorithm. The new stemmer contains four classes and each class contains several morphological conditions. The performance of new stemmer is examined with two grouped words. Rubam et al. [6] discussed various methods of affix removal stemmer. Also, they are analyzed four stemming techniques such as porter, lovins, paice and Krovetz stemmer and its merits and demerits of affix removal stemmers.

Sandeep et al. [7] analyzed strength of affix removal stemmers. Also, they are discussed comparative analysis of affix removal stemming algorithm accuracies. All tends to produce both over-stemming and under-stemming. Giridhar et al. [8] conducted a prospective study of stemming techniques in web documents. Prajensit et al. [9] is explained Yet Another Suffix Stripper (YASS) methods. YASS is difficult to decide a threshold for creating clusters and requires significant computing power. Venkat sudhakara reddy et al. [10] discussed stemming techniques applied to information extraction using RDBMS. SP. Ruba et al. [11] proposed APOST for increasing the performance of stemming. The performance of APOST stemmer examined with sample vocabulary downloaded from the web site http://snowball.tartarus.org/algorithms/english /voc.txt. It contains distinct words, arranged into "conflation groups". Some of them are incorrect words.

Rule-based stemmers ◦English - Lovins: originally proposed in 1968 (Lovins, 1968), this stemmer removes endings based on the longest-match principle. It uses a large list of endings, each of which is associated with one of a number of qualitative contextual restrictions that prevent the removal of endings in certain circumstances. It was the first stemmer to be published 1 . - Porter: published in 1980 (Porter, 1980), the Porter stemmer is the most widely used stemmer for the English language and has a series of suffix stripping rules. It has already been adapted to many languages under the Snowball framework. 2 - Paice/Husk: proposed in Paice (1990), this stemmer uses a table of rules. Each rule may specify the removal or replace- ment of an ending and the rules are grouped into sections corresponding to the final letter of the suffix. They are found the assumption that there would be a strong negative correlation between the errors made by the stemmer and the measures for IR performance such as MAP does not hold. The correlation between ERRT and MAP is weak for all four languages.

III. PROPOSED WORK

Enhanced Context Aware Stemming algorithm reduce different morphological modifications to their fundamental rules. Rule-based stemmers transform the variant word forms into their stems or base forms by using certain pre-defined language-specific rules. Hence, these are also called language specific stemmers. The creation of language-specific rules requires expertise in language or at least a native speaker of the particular language. Moreover, rule-based stemmers sometimes employ additional linguistic resources like dictionaries to conflate morphologically related words. Stemming is used to enable matching of queries and documents in keyword-based IR systems. This assumes that morphological variants of words have similar semantic interpretations and can be considered as equivalent for the purpose of IR applications. The Enhanced Context Aware stemmer rectifies the drawbacks of porter algorithm.

Description of ECAS

1. Select relevant text ending:
   - Examine the final letters of the

 DOI : 10.18231/2454-9150.2018.1341

document keyword;

2. Check applicability of rule:
   - If the final letters of the document keyword (DKW) match the ending rule1, Otherwise goto 3;
   - Delete from the right end of the DKW the number of characters specified by the remove rules;
   - if "add string", then append it to the end of the DKW;
   - if "replace string", then replace the number specified to the end of the DKW;
3. . Search for another rule
   - Go to the next rule in the rule engine database;
   - if the endings of the query term has changed, output stem, then terminate;
   - Otherwise goto 4.
4. Extract from Table (EFT)

EFT stemmed words match with document keywords and find correct stemmed word

5. Termination Condition

If matching endings acceptability conditions are satisfied, and then terminate the stemming process

6. If the stemmed word is meaningless goto 4 EFT method.
7. Output the stemmed word

Calculate (CSF) $= \frac{\text{Number of words correctly Stemmed}}{\text{Total number of words}}$

IV. Experimental Results

To evaluate the performance of the stemmer described in this thesis, we have applied these algorithms to the sample vocabulary downloaded from the web site http://snowball.tartarus.org/algorithms/english /voc.txt. It contains distinct words, arranged into "conflation groups". Some of them are incorrect words. For example, there are 155 **incorrect words** in the sample of 500 words which begin with alphabet 'b'.

To measure the strength and accuracy of stemmer, we considered a sample of 500 words containing 'b' alphabet words and analyze the result using the measuring criteria specified. This shows that the strength of all the stemmers is strong and all are aggressive in nature. APOST algorithm produced high word stemmed factor. But APOST algorithm is more aggressive than others.

Thus the accuracy of correctly stemmed words and conflating variant words of same group to correct stem is good, but not satisfactory, in APOST algorithm than the earlier stemmers. Table 3.2 shows the results of Lovins, Porter and ECAS algorithm.

Table 3.2: Results of Lovins, Porter and ECAS algorithm.

Analysis of Stemmers	Lovins Stemmer	Porter Stemmer	ECAS Algorithm
Total Words(TW)	500	500	500
Number distinct words before stemming (N)	425	425	425
Number distinct words after stemming (S)	184	204	210
Number of words stemmed (WS)	367	336	380
Words Stemmed Factor (WSF)	**73.35**	**67.17**	**76.00**
Correctly Stemmed Words (CSW)	102	107	125
Incorrectly Stemmed Words (ISW)	265	229	255
Correctly Stemmed Words Factor (CSF)	**27.80**	**31.97**	**32.89**
Correct Words not Stemmed (CW)	57	12	10
Number of Distinct Words after Conflation (NWC)	127	192	200
Average Words Conflation Factor	**24.8**	**25.52**	**28.0**

4.2.1. Word Stemmed Factor

Word Stemmed Factor obtains 73.35 by Lovins, 67.17 by Porter and 76% by APOST algorithm. Fig.3.5 shows comparison of words stemmed factor. The APOST stemmer performance is better than another existing stemming techniques.

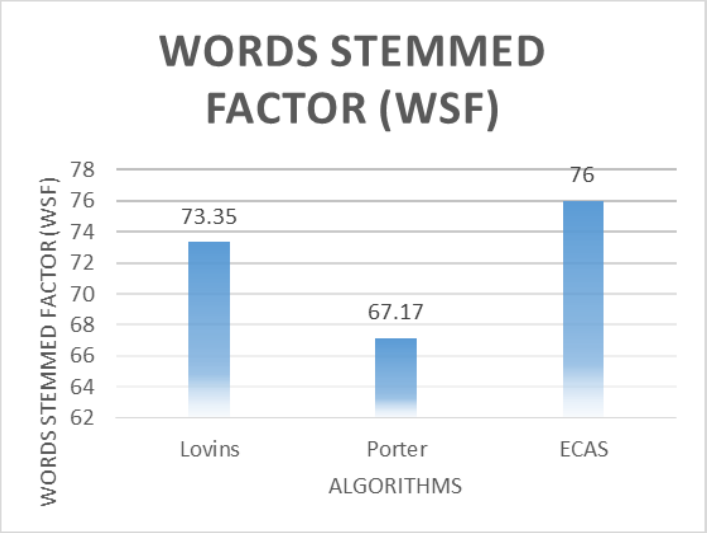

Fig.3.5: Comparison of Word Stemmed Factor.

4.2.2. Correctly Stemmed Word Factor

Correctly Stemmed Word Factor (CSWF) obtains 27.8% by Lovins, 31.97 by Porter and 32.89 by APOST algorithm. Fig.3.6 shows comparison of correctly stemmed word factor. The APOST stemmer produced more CSWF comparing to another stemmers.

 DOI : 10.18231/2454-9150.2018.1341

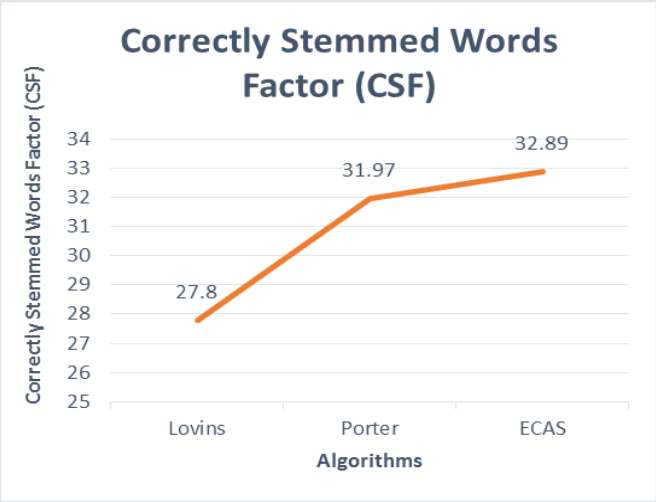

Fig.3.6: Comparison of Correctly Stemmed Words Factor.

4.2.3. Average Word Conflation Factor

Average Word Conflation Factor(AWCF) obtains -24.8% by Lovins, -8.52 by Porter and -28% by APOST algorithm. Fig.3.7 shows the comparison of Average Words Conflation Factor. The APOST stemmer is increases the word conflation factor.

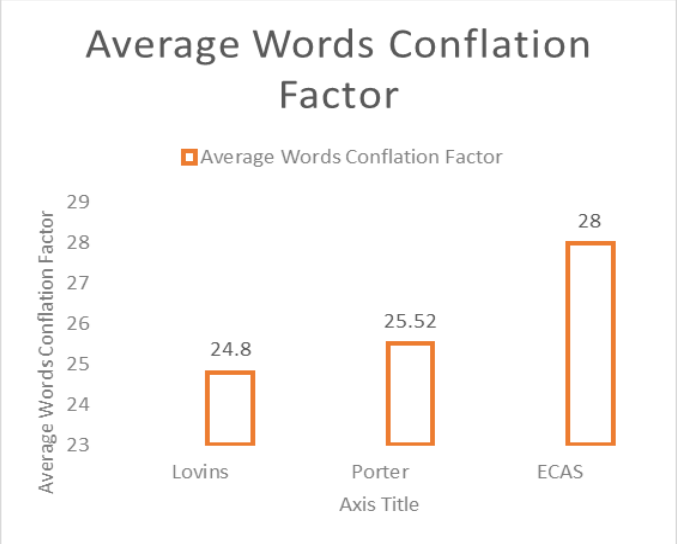

Fig.3.7: Comparison of Average Words Conflation Factor.

Further, the following things have been observed:

The word stemmed factor (WSF) obtained by APOST algorithm stemmer is comparatively high (76%), but CSF (32.89%) and AWCF (-29%) is comparatively low. This is because, besides the words having inflectional and derivational suffixes, it also transforms the root words to incorrect stem. This results in occurrence of both over-stemming and under-stemming errors. However, occurrence of over-stemming errors is more than under-stemming errors.

V. CONCLUSION

Stemming can be effectively used in Natural Language Processing. The benefits of stemming algorithm in text mining will be reduce the database size using identification of root word. Stemming techniques are useful in several fields such as Library and Information Science, Bio-Medical, Textual Database for the purpose of classification and indexing. The word stemmed factor (WSF) obtained by ECAS stemmer is comparatively high (76%), but CSF (32.89%) and AWCF (-29%) is comparatively low. The performance of ECAS stemmer is better than another existing stemmers. In future, to reduce the time consumption of Efficient Stemmer and decrease the utilization of memory storage.

.References

[1] Paik, J. H., Mitra, M., Parui, S. K., & Järvelin, K. GRAS: An effective and efficient stemming algorithm for information retrieval. ACM Transactions on Information Systems, 29 (4), 19, 2011.

[2] Brajendra Singh Rajput and NilayKhare, "A survey of Stemming Algorithms for Information Retrieval", IOSR Journal of Computer Engineering, Volume 17, Issue.3, pp. 76-80, 2015.

[3] S.P.Ruba Rani, B.Ramesh, M.Anusha, and J.G.R.Sathiaseelan, "Evaluation of Stemming Techniques for Text Classification" International Journal of Computer Science and Mobile Computing, Volume. 4, Issue. 3, pg.165 – 171 2015.

[4] Stefano Ferilli, Floriana Esposito and Domenico Grieco, "Automatic Learning of Lingustic Resources for Stop word Removal and Stemming from Text", Procedia Computer Science, Volume 38, pp.116-123, Elsevier, 2014.

[5] Wahiba Ben AbdessalemKaraa, "A new stemmer to improve information retrieval", International Journal of Network Security & Its Applications, Volume 5, No.4, pp. 143-153, July 2013.

[6] Rupan Gupta and Anjali Ganesh Jivani, "Empirical Analysis of Affix Removal Stemmers", International Journal of Computer Technology & Applications, Volume 5, Issue 2, pp. 393-399, March- April 2014.

[7] Sandeep R.Sirsat, Vinay Chavan and Hemant S.Mahalle, "Strength and Accuracy Analysis of Affix Removal Stemming Algotithms", International Journal of Computer Science and Information Technologies, Volume 4, Issue 2, pp.265-269, 2013.

[8] Giridhar N.S, Prema K.V and N.V Subba Reddy, "A Prospective Study of Stemming Algorithms for Web Text Mining", Ganpat University Journal of Engineering & Technology, Volume 1, Issue 1, pp. 28-34, Jan-Jun-2011.

[9] Prasenjit Majumder, MandarMitra, Swapan K. Parui, GobindaKole, PabitraMitra and Kalyankumar Datta, "YASS: Yet another suffix stripper", ACM Transactions on Information Systems, Volume 25, Issue 4, Article No. 18. 2007.

[10] Venkat Sudhakara Reddy. Ch and Hemavathi. D, "Information extraction using RDBMS and stemming algorithm", International Journal of Science and Research, Volume 3, Issue 4, pp. 503-507, April 2014.

[11] S.P. Ruba Rani, B.Ramesh and Dr.J.G.R.Sathiaseelan, "An Increasing Efficiency of Pre-processing using APOST Stemmer Algorithm for Information Retrieval", Journal of Emerging Technologies and Innovative Research, Volume 2, Issue 7, pp.3219-3223, July 2015.

 DOI : 10.18231/2454-9150.2018.1341