An Increasing Efficiency of Pre-processing using APOST Stemmer Algorithm for Information Retrieval

[1]S.P. Ruba Rani,[2]B.Ramesh,[3]Dr.J.G.R.Sathiaseelan
[1]M.Phil. Research Scholar,[2]Ph.D. Research Scholar,[3] Head, Department of Computer Science
[1,2,3]Department of Computer Science,
[1,2,3]Bishop Heber College, Tiruchirappalli, TN, India.

Abstract—**In the recent years, Text mining is an emerging research topic in Data Mining. Stemming is to find stem of particular word. Stemming technique is used to reduce words length to their origin form, by removing derivational and inflectional affixes. In this paper, we are proposed APOST (Advanced Porter STemming) algorithm for improving the efficiency of pre-processing in text mining. The APOST algorithm is enhanced version of porter stemmer. The APOST algorithm performance is compared with several algorithms. The APOST algorithm is consuming less time and memory space. Also, APOST is providing more accuracy and less error rate.**

IndexTerms—**Text Mining, Pre-processing, Stemming Techniques, APOST Algorithm, Porter Stemming, Lovins Stemmer.**

I. INTRODUCTION

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 stemming is to standardize words by reducing a word to its base. Data mining is a process of discovering hidden patterns and information from the existing data. Data mining [1] techniques are very useful to manipulating and analyzing data from database. They are several techniques are available in data mining for analyzing data such as clustering, classification, decision tree, neural network and genetic algorithm. Among all these types of data [2], particularly data mining supports text data for representing the document. A document consists of collection of words which includes stop words. Many words used in the text are morphological variants which based from the root form e.g. connection /connect, combining /combine, preferences /preferred/prefer. Text mining is a new and exciting research area that tries to solve the information overload problem by using techniques from data mining, machine learning, natural language processing (NLP), information retrieval (IR), and knowledge management.

Text classification [3] is one of the major parts in text mining. Nowadays, handling textual documents is a great challenge. Text classification uses several key classification algorithms, e.g., decision trees, pattern (rule)-based classifiers, support vector machines, naïve Bayesian classifier and artificial neural networks. B.Ramesh et al. [4] elaborately discussed several key pre-processing techniques. Text classification applied in many fields. Biological genetic algorithm for instance selection of text classification in medical field [5]. Brajendra et al. [6] analyzed several recent stemming techniques and provided several directions. Ruba et al. [7] explained various stemming methods and their constraints. Ramesh et al. [8] discussed several instance selection methods in pre-processing. Instance selection is ananother solution of text pre-processing.

II. LITERATURE REVIEW

The enormous amount of information stored in unstructured texts cannot simply be used for further processing by computers, which typically handle text as simple sequences of character strings. Therefore, specific preprocessing methods and algorithms are required in order to extract useful patterns. 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. Stefano et al. [9] discussed automatic learning methods of linguistic resources for stop words removal. Wahiba et al. [10] proposed new stemmer for rectifying the limitations of porter stemmer algorithm. The new stemmer contains four classes and each class contains several morphological conditions. Ruban et al.[11] discussed various methods of affix removal stemmer. They are analyzed merits and demerits of affix removal stemmers.

Sandeep et al. [12] analyzed strength of affix removal stemmers. Also, they are discussed comparative analysis of affix removal stemming algorithm accuracies. Giridhar et al. [13] conducted a prospective study of stemming techniques in web documents. Prajensit et al. [14] is explained Yet Another Suffix Stripper (YASS) methods. YASS is difficult to decide a threshold for creating clusters and requires significant computing power. Venkatsudhakarareddy et al. [15] discussed stemming techniques applied to information extraction using RDBMS.

III. Proposed Work

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. Fig.1. shows the overview of proposed system. The APOST stemmer rectifies the drawbacks of porter algorithm.

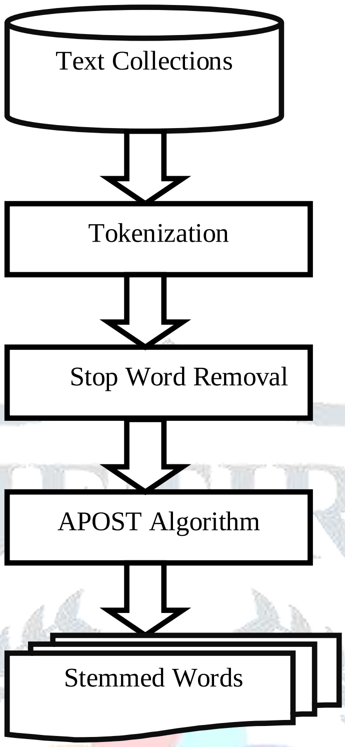

Fig.1. overview of proposed system.

APOST ALGORITHM DESCRIPTION

In this section, thesis describes step by step method of our APOST algorithm as follows:

Step 1 Initialization:
Input the text collection.
Step 2 Select relevant text ending:
Examine the final letters of the text collection;
Consider the first rule in the relevant ending for the input query term, and to indicate the first query term among stemming candidates.
Step 3 Check applicability of rule:
If the final letters of the query term do not match the ending rule, output stem, then terminate;
if the final letters of the query term and the ending rule matches, then goto 4;
if the final letters of the query term and the rule matches, and matching ending acceptability conditions are not satisfied, then goto 5;
Step 4 Apply rule:
Delete from the right end of the token the number of characters specified by the remove rules;
if there is an add string, then add it to the end of the query term specified by the rule;
if there is a replace string, then replace the number specified to the end of the query term;
if the condition specified is "no applicable rule" output the stem, then terminate;
if the condition specified is "match ending found" then take output to the next rule to access;
Otherwise goto 2.
Step 5 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 3.
Step 6 Termination Condition:
If matching endings acceptability conditions are satisfied, and then terminate the stemming process.

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. The APOST stemmer is developed using HTML and PHP with java script. For example, there are 155incorrect wordsin 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. The result of the most noticeable aggressive stemmers referred in this thesis is shown in Table 1.

From Table 1, it is observed that theword stemmed factor (WSF) obtained by all the algorithms are above 65% which is above the threshold value. 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. However, there is comparatively large difference between Lovins stemmer and the other stemmers on AWCF, but not differs much on CSWF as shown in table 2.

Thus the accuracy of correctly stemmed words and conflating variant words of same group to correct stem is better in APOST algorithm than the earlier stemmers.

Table1. Comparison of Algorithm Results

Analysis of Stemmers	**Lovins Stemmer**	**Porter Stemmer**	**APOST 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	4	10
Number of Distinct Words after Conflation (NWC)	127	200	200
Average Words Conflation Factor	**24.8**	**8.52**	**28.0**

Word Stemmed Factor

Word Stemmed Factor obtains 73.35 byLovins, 67.17 by Porter and 76 by APOST algorithm. Fig.2. shows comparison of words stemmed factor. . The APOST stemmer performance is better than another existing stemming techniques.

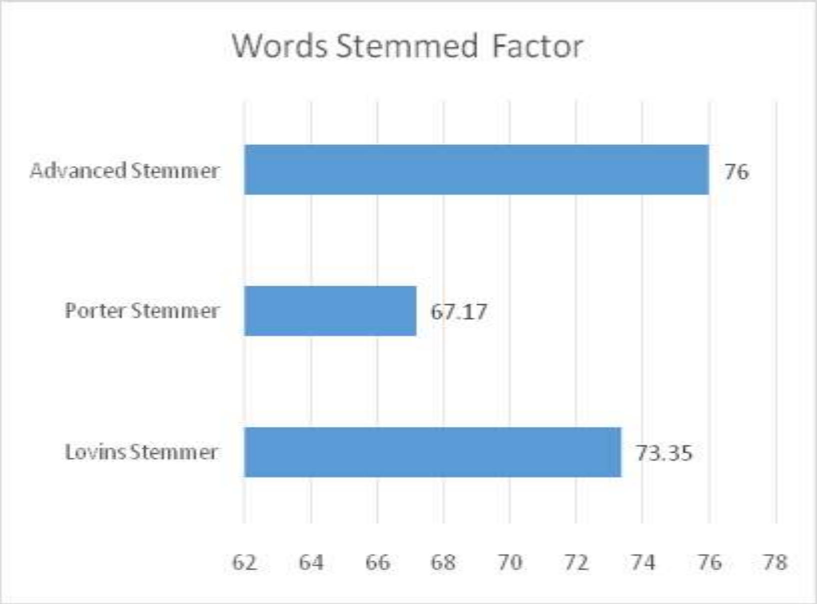

Fig.2. Comparison of Word Stemmed Factor.

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. shows comparison of correctly stemmed word factor. The APOST stemmer produced more CSWF comparing to another stemmers.

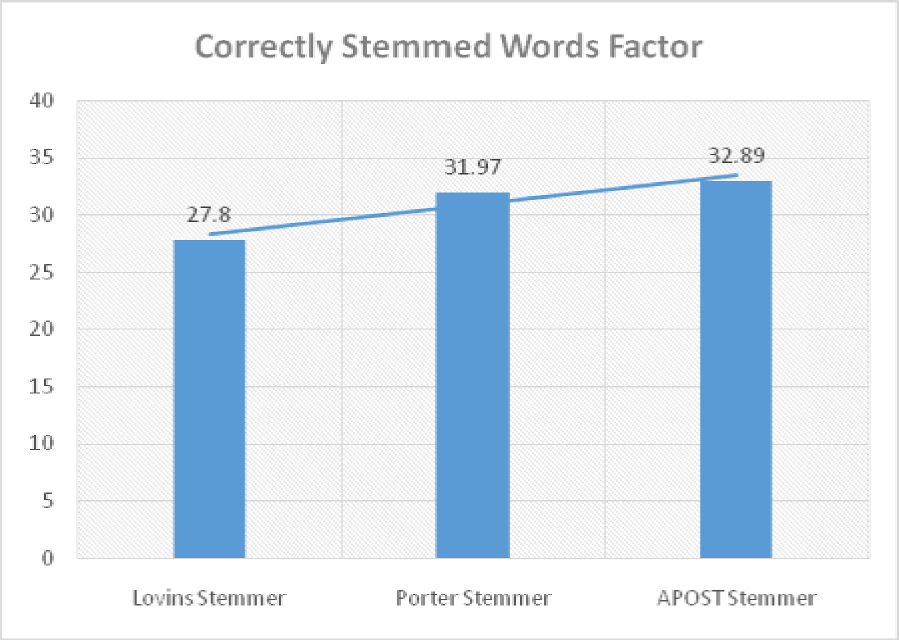

Fig.3. Comparison of Correctly Stemmed Words Factor.

Average Word Conflation Factor

Average Word Conflation Factor (AWCF) obtains -24.8 by Lovins, 8.52 by Porter and 28 by APOST algorithm. Fig.4. shows the comparison of Average Words Conflation Factor. The APOST stemmer is increases the word conflation factor.

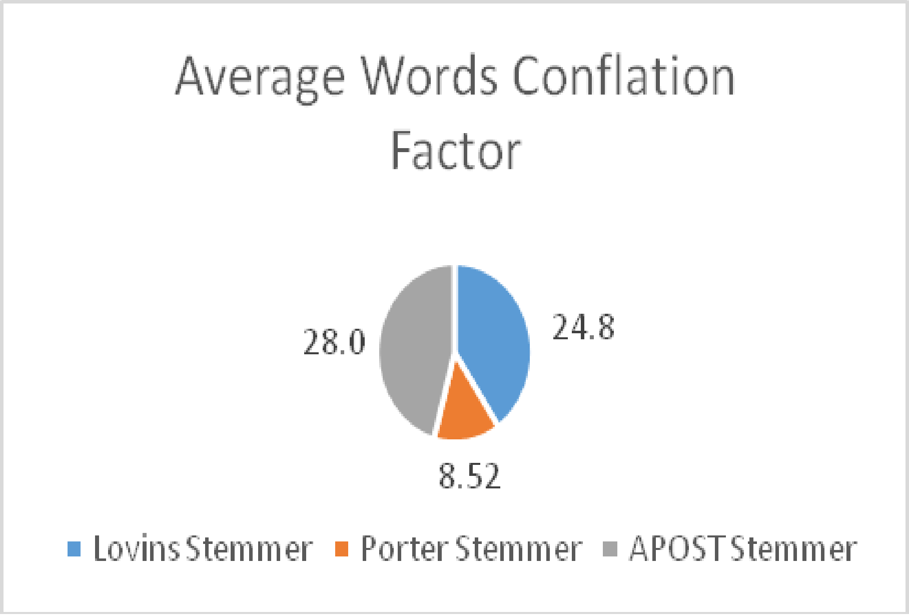

Fig.4. Comparison of Average Words Conflation Factor.

V. Conclusion

Stemming can be effectively used in natural language processing such as in free text search. The use of stemming algorithms before mining will reduce the database size. Stemming is useful for library and information science professional in the fields of classification and indexing, as it makes the operation less dependent on particular forms of words. Hence, none of the stemming algorithms give 100% output but is good enough to be applied to the text mining, NLP or IR applications. The APOST stemmer obtains 76 word stemmed factor, 32.89 correctly word stemmed factor and 28 average word coflation factor. The APOST stemmer is produced less error rate and more conflated words. The performance of APOST stemmer is better than another existing stemmers. In future, to reduce the time consumption of APOST Stemmer and decrease the utilization of memory storage.

REFERENCES

[1] M.S.B. PhridviRaj and C.V. GuruRao, “Data mining – past, present and future – a typical survey on data streams”, Elsevier, 2013.

[2] R. Sagayam, S.Srinivasan and S. Roshni., " A Survey of Text Mining: Retrieval, Extraction and Indexing Techniques”, International Journal of Computational Engineering Research (ijceronline.com) Vol. 2 Issue. 5, September 2012.

[3] Francesco Colace, Massimo De Santo, Luca Greco and Paolo Napoletano, “Text classification using a few labeled examples”, Computer in Human Behavior 30(2014)689-697, Elsevier, 2013.

[4] B.Ramesh, J.G.R.Sathiaseelan, “A Theoretical Stuudy on Advanced Techniques in Pre-Processing and Text Classification” International Journal of Data Mining and Emerging Technologies, Vol.5, No.1, pp.6-10, 2015.

[5] AlperKursatUysal and SerkanGunal, “The impact of preprocessing on text classification”, Information Processing and Management 50(2014) 104-112, Elsevier, 2013.

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

[7] 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, *Vol. 4, Issue. 3, pg.165 – 171 2015.*

[8] B.Ramesh, J.G.R.Sathiaseelan, “An Analysis of Instance Selection Algorithms Using Support Vector Machine for Text Classification” International Journal of Modern Computer Science, Vol.3, Issue.2, pp.81-84, 2015.

[9] Stefano Ferilli, Floriana Esposito and Domenico Grieco, “Automatic Learning of Lingustic Resources for Stopword Removal and Stemming from Text”, Procedia Computer Science 38 (2014) 116-123, Elsevier, 2014.

[10] Wahiba Ben AbdessalemKaraa,”A new stemmer to improve information retrieval”, International Journal of Network Security & Its Applications, July 2013.

[11] Rupan Gupta and Anjali Ganesh Jivani, “Empirical Analysis of Affix Removal Stemmers”, IJCTA, March- April 2014.

[12] 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, Vol. 4(2), 2013, 265-269.

[13] GiridharN.S,Prema K.V and N.V SubbaReddy,”A Prospective Study of Stemming Algorithms for Web Text Mining”, Ganpat University Journal of Engineering & Technology,Vol-1,Issue-1,Jan-Jun-2011.

[14] PrasenjitMajumder, MandarMitra, Swapan K. Parui, GobindaKole, PabitraMitra and KalyankumarDatta. “YASS: Yet another suffix stripper”. ACM Transactions on Information Systems. Volume 25, Issue 4. 2007, Article No. 18.

[15] VenkatSudhakaraReddy.Ch and Hemavathi.D, “Information extraction using RDBMS and stemming algorithm”, International Journal of Science and Research (IJSR), April 2014.