Volume 3, Issue 7, July 2013 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Improvised Stemming Algorithm - TWIG

S.Santhana Megala* *Research Scholar, PRIST University Thanjavur, Tamil Nadu, India*
Dr.A.Kavitha *Dept. of Computer Science Kongunadu Arts & Sci College Coimbatore, Tamil Nadu, India*
Dr. A.Marimuthu *Department of Computer Science, Govt. Arts College Coimbatore , Tamil Nadu, India*

Abstract— Information Retrieval System retrieves the document from a collection of documents which is need by the user and it should satisfy the users need also. IR System makes use of Stemming algorithms in the pre-processing stage to convert the words to their root form, which improves the retrieval performance of the system. Many such stemming algorithms exist; among those porters stemming algorithm is the popular one because of its simplicity, efficiency and availability. Even It is popular there exist some drawbacks. In this paper a TWIG stemming algorithm is proposed which produced meaning full stems and it reduces the error rate to a bare minimum one.

Keywords—Information Retrieval, Preprocessing, Stemming, Root Word, Suffix Removal, Legal Text Summarization.

I. INTRODUCTION

Information Retrieval is a process of finding relevant information from the resources which satisfy the user's need, which is done by the support of indexing and searching systems. These systems extract the index terms or keywords from the text present in the source documents. Word Stemming is the main aspect supported by indexing and search systems, to improve the recall by reducing the words to their root words at the time of indexing and searching. Thus a Word Stemming is needed to improve the performance of an Information Retrieval system. Word Stemming is a method of reducing the words to their root word by removing the attached suffixes and prefixes before indexing, to combine the words 'used', 'using', 'uses' to the word 'use'. It simply reduces the grammatical or inflectional or derivational form of a word to a common base form called stem. This paper set in motion with the brief work done by porter for stemming in section 2, and its drawbacks are explained in the section 3. Section 4 introduces the enhanced work for stemming algorithm which is proposed. The Experiments and Results are covered in section 5. Finally the conclusions and plan for future work is discussed in section 6.

II. CLASSIFICATION OF STEMMING

Stemming Algorithms are broadly classified into three groups, based on the archetypical way of finding the stem from the word. They are (1) Truncating (2) Statistical (3) Mixed.

A. Truncating

From the name itself it says that," It is a method of removing the Affixes (Suffixes or Prefixes) of a word. It truncated a word at nth position keeping the n letters and removes the rest based on some rules and conditions. This type of stemmer is the most basic one as well as a popular one.

B. Statistical

In this method word Stemming is done after applying certain Statistical Techniques like N-Gram, HMM, YASS. This type of stemmers is based on statistical Analysis and techniques.

C. Mixed Method

This method involves in the combination of Inflectional (Plural, Gender) & Derivational morphological Analysis methods (POS), Corpus Based and Context Sensitive. Hence this paper followed the simple and efficient truncating algorithm for stemming. Many such stemming algorithms are exist; among them Porters Stemming Algorithm is selected as a base, and by adding some rules from STANS algorithm and some slight improvements is made to increase the accuracy as well as to get the meaning full stem in the pre-processing stage.

III. PORTER'S STEMMING ALGORITHM

In 1980, at the University of Cambridge, Martin Porter Developed the Porter Stemming Algorithm [1]. Porter's Stemming Algorithm is a conflation (act of fusing or combining) Stemming Algorithm. It is a suffix removal algorithm. As defined in [1] "It is a method of removing the common morphological and inflexional endings words in English. Stemming occupies an important part in term normalization process, when setting up IR Systems. Porter's algorithm looks for nearly 60 suffixes (60 rules). It removes the smaller and simpler suffixes attached to the stem and if the stem is too short, it does not remove the suffix. Porter's algorithm has five steps and in each step certain rules are applied. It calculates the words based on the Vowel (A, E, I, O, U) Consonant (Other than Vowel) Pairs, and it is denoted by 'm'. By using multiple step Process it successively removes the short suffixes, instead of removing a single longest possible suffix.

I.Draw Backs in Porter's Algorithm:

Even though the porter's algorithm is simple and efficient it also has some drawbacks such as most of the output stems are

meaningless (generalization → gener, etc...) and it is suitable for American English but we follow the British English [5]. Ex: American English does not keeps silent e at the ending but British English keeps it. (Before→Befor) Ex: American English ends with I but British English ends mostly with 'y' (Verify → verifi, etc..) and it cause Over-stemming Problem (Distinct words may wrongly conflate to give same stem Ex: Paste, Past → Past) and Under-stemming Problem (same words remain distinct after stemming). Among these, over stemming cause serious problems in performance of IR Systems while under stemming does not affect the IR System. Long and Complex suffixes are truncated letter by letter by using different steps, finally which leads some stems to meaningless. This algorithm mainly focuses on the speed and simplicity but it shows the absence of lexicon (dictionary) results which cause improper conflation and loss of precision.

IV. STANS ALGORITHM

STANS Algorithm is obtained by modifying the porters stemming algorithm. In this algorithm totally 31 modifications is done among that 15 rules were added, 11 rules were modifies and 5 rules were deleted from the origin. Now the STANS Algorithm (modified Porters algorithm) has 65 rules to truncate the suffixes. Those changes are listed in the table1. STANS Algorithm is applied in the previous version of porters stemming algorithm.

TABLE I
STANS ALGORITHM
(Modification done in STANS Algorithm)

Step No.	Rule No.	Operation	Porter's algorithm	STANS algorithm
1	1	Add	--	US→ US
	2*	Add	--	CEED→ CESS
	3*	Add	--	EED→ EED
	4	Modify	IES→I	IES→Y
	5	Add		IED→Y
	6	Modify	(*V*)ING→	(*V*)ING→ E
	7	Delete	(*V*)Y→I	
	8	Add		ED→ E
2	9	Modify	ANCI→ANCE	ANCY→ ANCE
	10	Modify	ENCI→ENCE	ENCY→ ENCY
	11	Modify	ABLI→ABLE	ABLY→ABLY
	12*	Modify	OUSLI→OUS	OUSLY→ OUS
	13*	Modify	ALITI→AL	ALITY→ AL
	14*	Modify	IVITI→IVE	IVITY→IVE
	15*	Modify	BILITI→BLE	BILITY→ BLE
	16	Add		FULLY→FUL
	17*	Add		FUL→
	18	Add		LESSLY→ LESS
	19	Add		BLY→BLE
3	20	Add		LESS→
	21*	Modify	ICITI→IC	ICITY→ IC
4	22	Modify	AL→	AL→ E
	23	Add		IABLE→ Y
	24	Delete	ANCE→	
	25	Add		SCOPIC→ SCOPE
	26	Delete	IC→	
	27	Delete	ATE→	
	28*	Add		FYE→ FY
	29*	Add		ALLY→AL
	30*	Add		TLY→T
5	31	Delete	E→	

In Step: 1 totally 8 modifications is done, which are Adding US→US for words like Serious, Various etc.. & CEED→ CESS for words like Succeed as success, & eed→eed to keep the words greed, deed, & IED→Y for Modified →Modify & ED→E for provided to provide. & IES→Y for ponnies to ponni & Modify (*v*) ING→ as (*v*) ING→ E for Scoring to score. In Step: 2 (*v*)Y→I is deleted to avoid verify as verifi. In Step: 3 10 changes are done to add FULLY→FUL for usefully→ to useful & LESSLY → LESS for carelessly as careless & BLY→BLE for possibly to possible & all the 'i' is modified as 'y' in the following rules ANCI, ENCI, ABLI, OUSLI, ALITI, IVITI, BILITI. In Step: 4 ICITI is modified as ICITY and LESS→ is added for worthless to worth.

In Step: 5 the following rules are added FYE→FY for Purifying as Purifye as Purify & ALLY→ AL for finally as final & TLY→T for Significantly to significant & IABLE →Y for Simplifiable to Simplify & SCOPIC →SCOPE for Microscopic as Microscope & Modify AL→ to AL→E for revival to revive & delete IC→ & ATE→ . In Step: 6 Eliminating E→ is omitted to avoid deleting 'e' from provide.

- *Drawbacks in STANS*

In Step: 3 by using the rule LESS→ the meaning of the word is changed to opposite one (Careless will change to care, which cause errors) & FUL→ is already used by porter in step 4. And beyond that there are still many words, which have no meaning.

V. IMPROVISED STEMMING ALGORITHM -TWIG

Without violating the simplicity of Porters Stemming Algorithm, 31 modifications is done in the STANS Algorithm, among that 29 Modifications are adapted to the improvised stemming algorithm except (LESS→ , FUL→) and 15 new rules are introduced to improve the performance of the IR System with meaning full stems.

TABLE 2
TWIG ALGORITHM

Step No.	Rule No.	Operation	Porter's algorithm	STANS algorithm
1	1	Add	--	US→ US
	2*	Add	--	CEED→ CESS
	3*	Add	--	EED→ EED
	4	Modify	IES→I	IES→Y
	5	Add		IED→Y
	6	Modify	(*V*)ING→	(*V*)ING→ E
	7	Delete	(*V*)Y→I	
	8	Add		ED→ E
2	9	Modify	ANCI→ANCE	ANCY→ ANCE
	10	Modify	ENCI→ENCE	ENCY→ ENCY
	11	Modify	ABLI→ABLE	ABLY→ABLY
	12*	Modify	OUSLI→OUS	OUSLY→ OUS
	13*	Modify	ALITI→AL	ALITY→ AL
	14*	Modify	IVITI→IVE	IVITY→IVE
	15*	Modify	BILITI→BLE	BILITY→ BLE
	16	Add		FULLY→FUL
	17*	Add		FUL→
	18	Add		LESSLY→ LESS
	19	Add		BLY→BLE
3	20	Add		LESS→
	21*	Modify	ICITI→IC	ICITY→ IC
4	22	Modify	AL→	AL→ E
	23	Add		IABLE→ Y
	24	Delete	ANCE→	
	25	Add		SCOPIC→ SCOPE
	26	Delete	IC→	
	27	Delete	ATE→	
	28*	Add		FYE→ FY
	29*	Add		ALLY→AL
	30*	Add		TLY→T
5	31	Delete	E→	

In the Improvised Stemming algorithm, 11 new rules are added to the Step 1. Which are (if ((*V*) ED|ING)) then I→Y, to display the Words like Modified as Modify not as Modif, and AS→ASE, to display the Words like Erased as Erase not as Eras, and DG→DGE, to display the Words like Acknowledged as Acknowledge not as Acknowledg, and LV→LVE, to display the Words like Dissolved as Dissolve not as Dissolv, and US→USE, to display the Words like Caused as Cause not as Caus, and AC→ACE, to display the Words like Displaced as Displace not as Displac, and AG→AGE, to display the words like Encouraged as Encourage not as Encourag, and AS→AS to display the words like Has ad Has not as ha, and IS→IS to display the words like His as His not as Hi, and ER→ER to display the Words like Lawyer as Lawyer not as Lawy, and INGLY→ is used to display the words like Disturbingly as Disturb. In Step 2 as described in STANS Algorithm (*V*) Y→I is deleted. In Step 3 the Changes did in STANS Algorithm is adopted. In Step 4 the rule LESS→ is deleted because it gives the negative meaning like Useless as Use, Helpless as Help. In Step 5 the rule AL→E is modified to AL→AL to display the words the Legal as legal and INANT→ INANT is added to display the words as Complainant as Complainant not to Complai and IDENT → IDENT is added to display the word President as President not as Presid and ISE→ is added to display the word Authorize to Author. In Step 6 the Changes did in STANS Algorithm is adopted.

VI. EXPERIMENTS AND RESULTS

The Performance of the Proposed Stemming Algorithm -TWIG is measured based on the No. of Meaning full words (Stems) it generated, Which is calculated by finding the proportion of the Unmeaning full words with the Meaning full words generated and the experiment output is showed based on the error rate producing by the Stemming Algorithms and a comparison is made on that. The Inputs used for the experiment is Alan Beale's Core Vocabulary Dictionary data, which contain 21,877 Words. Thus the error rate of the proposed algorithm - TWIG is 1.5%, which is a bare minimum one, when compare with the 7.6% by STANS and 39.9% by Porters Stemming Algorithm.

Even though the size of the output is slightly larger than the Porter's and Stans algorithm it is negligible when compared to the Meaningful output, because if the stem word is not a meaningful Word in the Preprocessing stage then it need a manual correction at the post processing this leads a time delay and need man power to execute the process. To avoid such situation and to bring the fully automatic concept, the proposed TWIG algorithm is implemented in the preprocessing stage.

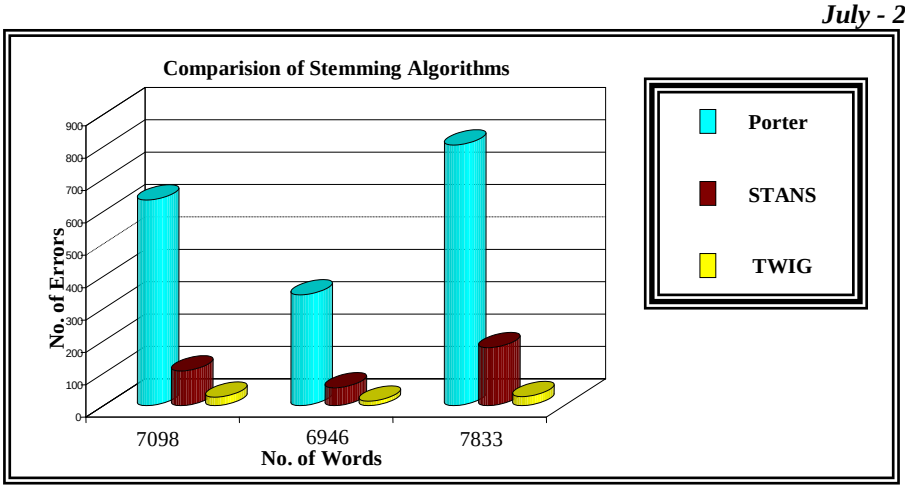

Fig. 1 Comparision of Stemming Algorithms

VII. CONCLUSIONS AND FUTURE WORK

In this paper an improvised stemming algorithm is introduced to produce a clear and meaningful stem, which is based on the famous Porters stemming algorithm. A Slight modification is done without compromising the efficiency and simplicity of porter's algorithm. The experimental result shows that the improvised algorithm shows a better accuracy in generating a meaning full stems comparing to standard porters algorithm, which reduces the error rate (Over stemming and Under stemming) to a bare minimum one. The improvised stemming algorithm is further applied to the research work on summarization and classification of textual data in future to utilize the efficiency and simplicity.

REFERENCES

[1] M.F. Porter, "An Algorithm for Suffix Stripping", Program, 14(3), P: 130-137, 1980.
[2] M.F. Porter, "Developing the English Stemmer", http://snowball.tartarus.org/., 2002.
[3] Anjali Ganesh Jivani, "A Comparative Study of Stemming Algorithms" International Journal of Computer Technology and Application, P: 1930 - 1938, 2011.
[4] S. Srinivasan and P. Thambidurai, "STANS Algorithm for Root Word Stemming", Information Technology Journal, P: 685 – 688, 2006.
[5] Http://en.wikipedia.org/wiki/American_and_British_English_Spelling_differences#Historical_origins
[6] Fadi Yamout, Rana Demachkieh, Ghalia Hamdan, and Reem Sabra, "Further Enhancement to the Porter's Stemming Algorithm", Machine Learning and Interaction for Text based Information Retrieval, P: 7- 23, 2004.
[7] Alan Beale's Core Vocabulary Compiled form 3 Small ESL Dictionaries (21, 877) This is relase 4.0 of 12dicts, released Jan. 18, 2003.
[8] S. Santhana Megala, A. Marimuthu, "A Study on Text Summarization Techniques and its Applications", National Conference on Recent Trends and Advances in Information Technology, 2012, P: 5.
[9] S. Santhana Megala, A. Marimuthu, "A Comparative Analysis of Legal Text Summarization", International Conference on Design and Applications on Structures, Drives, Communicational and Computing Devices, 2012.
[10] Krishna Kumar Mohbey, Sachin Tiwari, Preprocessing and Morphological Analysis in Text Mining, International Journal of Electronics Communication and Computer Engineering, 2011.

- S.Santhana Megala is currently pursuing Ph.D in Computer Science in PRIST University, Thanjavur, Tamil Nadu, and working as an Assistant Professor in SNMV College of Arts & Science, Coimbatore, Tamil Nadu.
- Dr. A.Kavitha is currently working as an Assistant Professor in Computer Science, Kongunadu Arts and Science College, Coimbatore – 29, Tamil Nadu, India.
- Dr. A. Marimuthu is currently working as an Associate Professor in Government College of Arts & Science, Coimbatore, Tamil Nadu.

