IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Volume-3, Issue-8 E-ISSN: 2347-2693

Enhanced Suffix Stripping Algorithm to Improve Information Retrieval

Sundar Singh[1], R K Pateriya[2]

[1*,2] *Department of Computer Science & Engineering*
Maulana Azad National Institute of Technology Bhopal, India, 462003

www.ijcseonline.org

Received: Jul /11/2015 Revised: Jul/27/2015 Accepted: Aug/16/2015 Published: Aug/30/ 2015

***Abstract*—** Stemming algorithms are used to convert the words in text into their grammatical base form, and are mainly used to increase the Information Retrieval System's efficiency. Several algorithms exist with altered techniques. The most widely used is the Porter Stemming algorithm. However, it still has several drawbacks, although many attempts were made to improve its structure. This paper discloses the inaccuracies encountered during the stemming process and proposes the corresponding solutions.

***Keywords*—** Stemming, stop word, Text mining, NLP, IR.

I. INTRODUCTION

Stemming is a technique used to reduce words to their root form called stem, by removing derivational and inflectional affixes. Most of the existing stemming algorithms uses affix stripping technique. This technique has wide application in NLP, Text mining and information retrieval. Stemming improves the performance of information retrieval systems by decreasing the index size. There are many stemming algorithms implemented for English language. Many of these algorithms are working successfully in information retrieval system. However there are many drawbacks in stemming algorithms, since these algorithms can't fully describe English morphology [1].

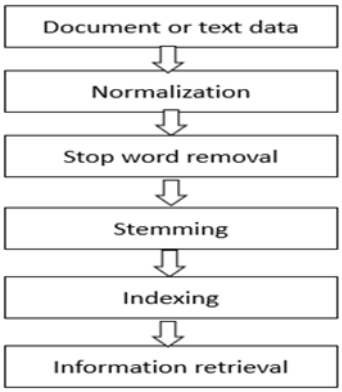

Fig. 1: Various steps in IR

Porter stemming algorithm is most widely used algorithm for English language .It is efficiently working in many informational retrieval systems.it also have some kind of errors like over-stemming and under-stemming because English morphology is very wide .By adding new rules in porter's algorithm will make it more efficient in the context of information retrieval. Enhanced suffix stripping algorithm is having the less amount of over-stemming and under-stemming errors with less amount of index size [2].

These are various steps in IR first upload a document or type a paragraph or text which is to be stemmed. After that normalize the text data such that it is converted into either lower case or upper case. All the special characters are removed in this stage. In the 3rd step remove stop words. Stop words are connecting words like is, are, the, am, be, etc. these words do not have their own meaning. In the 4th step stemming will be done, all words are converted into their root or base or stem form [3]. Then indexing of all the stems will be done. Stemming reduced the index size approximately 1/2 of its previous word count. Then information retrieval will be done.

II. LITERATURE REVIEW

A. *Porter's Algorithm*

Porter stemming algorithm is one of the most famous stemming method proposed by martin porter in 1980. It comprises 60 rules in five steps. It is based on suffix stripping technique. The data passes through different steps one by one, so it is a multi-pass algorithm. Many reformations and enhancements have been done and proposed on the basic algorithm. It is based on the idea that there are approx. 1200 suffixes in the English language, mostly made up of a grouping of smaller and multiple suffixes. The algorithm has five steps, and within each step, rules are applied until one of them passes the conditions. If a rule is matched, the suffix is removed consequently, and the next step is performed. The resultant stemmed word at the end of the fifth step is returned by the algorithm. Many version of porter stemming are released [4].

The rules look like the following pattern:
<Condition> <suffix> → <new suffix>
For example, a rule (m>0) EED → EE means "if the word has at least one "vowel and consonant" pair plus EED

ending, change the ending to EE". Example "agreed" is converted as "agree" whereas "feed" remains unaffected. This algorithm has about 60 rules in five steps. It is most widely used algorithm for the purpose of stemming but it has many errors in the rules. Several modifications have been done on porter algorithm but it still have over stemming and under stemming kind of problems.
Advantages of this algorithm are its efficiency and less time consumption. The main disadvantage of this algorithm is that it has over stemming and under stemming type of problems [5,6].

Stemming rules are represented by the single form.

[C](VC){m}[V].

Here 'm' will be called the measure, number of VC pairs of any word.
The rules for removing a suffix will be given in the form.

(Condition) S1 → S2

This means that if a word ends with the suffix S1, and the stem before S1 satisfies the given condition, S1 is replaced by S2. The condition is usually given in terms of m. and m is calculated by only condition part not S1.
The `condition' part may also contain the following:
*S - the stem ends with S (and similarly for the other letters).
v - the stem contains a vowel.
*d - the stem ends with a double consonant (e.g. -TT, -SS).
*o - the stem ends cvc, where the second C is not W, X or Y (e.g: -WIL, -HOP)[2].

In a set of rules written beneath each other, only one is obeyed, and this tills the one with the longest matching S1 for the given word. For example

sses →- ss
ies →i
ss →ss
s →

In the rules below, examples of their application, successful or otherwise, are given on the right in lower case. The algorithm now follows:

Step 1a

(Condition)S1→S2	Word	Stem
sses →ss	Caresses	caress
ies →i	Ponies	poni
	Ties	ti
ss → ss	Caress	caress
s →	Cats	cat

Step 1b

(Condition)S1→S2	Word	Stem
(m>0) eed → ee	Feed	feed
	Agreed	agree
(*v*) ed →	plastered	plaster
	bled	bled
(m>0)(*v*) ing →	Motoring	motor
	Sing	sing
at → ate	conflat(ed)	conflate
bl → ble	troubl(ed)	trouble
iz → ize	siz(ed)	size
	fall(ing)	fall
	hiss(ing)	hiss
(m=1 and *o) → E	fail(ing)	fail
	fil(ing)	file

The rule to map to a single letter causes the removal of one of the double letter pair. The -E is put back on -AT, -BL and -IZ, so that the suffixes -ATE, -BLE and -IZE can be recognized later. This E may be removed in step 4.

Step 1c

(Condition)S1→S2	Word	Stem
(*v*) Y -> I	happy	happi
	Sky	sky

Step 1, deals with plurals and past participles. The subsequent steps are straighter forward.

Step 2

(Condition)S1→S2	Word	Stem
(m>0) ational→ ate	relational	relate
(m>0) tional→ tion	conditional	condition
	rational	rational
(m>0) enci→ ence	valenci	valence
(m>0) anci→ ance	hesitanci	hesitance
(m>0) izer → ize	digitizer	digitize
(m>0) abli→ able	conformabli	conformable
(m>0) alli→ al	radicalli	radical
(m>0) entli→ ent	differentli	different
(m>0) eli→ e	vileli	vile
(m>0) ousli→ ous	analogousli	analogous
(m>0) ization→ ize	vietnamization	vietnamize
(m>0) ation→ ate	predication	predicate
(m>0) ator→ ate	operator	operate
(m>0) alism→ al	feudalism	feudal
(m>0) iveness→ ive	decisiveness	decisive
(m>0)fulness → ful	hopefulness	decisive
(m>0)ousness→ ous	callousness	callous
(m>0) aliti → al	formaliti	formal
(m>0) iviti → ive	sensitiviti	sensitive
(m>0) biliti → ble	sensibiliti	sensible

The test for the string S1 can be made fast by doing a program switch on the penultimate letter of the word being tested. This gives a fairly even breakdown of the possible values of the string S1. It will be seen in fact that the S1-strings in step 2 are presented here in the alphabetical order

of their penultimate letter. Similar techniques may be applied in the other steps.

Step 3

(Condition)S1→S2	Word	Stem
(m>0) icate→ ic	triplicate	triplic
(m>0) ative→	formative	form
(m>0) alize→al	formalize	formal
(m>0) icity→ic	electriciti	electric
(m>0) ical→ ic	electrical	electric
(m>0) ful→	hopeful	hope
(m>0) ness→	goodness	good

Step 4

(Condition)S1→S2	Word	Stem
(m>1) al→	revival	reviv
(m>1) ance→	allowance	allow
(m>1) ence→	inference	infer
(m>1) er→	airliner	airlin
(m>1) ic→	gyroscopic	gyroscop
(m>1) able→	Adjustable	adjust
(m>1) ible→	Defensible	defens
(m>1) ant→	Irritant	irrit
(m>1) ement→	adjustment	adjust
(m>1) ent→	Dependent	depend
((m>1 and (*s,*t))ion→	Adoption	adopt
(m>1) ou→	homologou	homolog
(m>1) ism→	communism	commun
(m>1) ate→	activate	activ
(m>1) iti→	Angularity	angular
(m>1) ous→	homologous	homolog
(m>1) ive→	Effective	effect
(m>1) ize→	bowdlerize	bowdler

Step 5a

(Condition)S1→S2	Word	Stem
(m>1) e→	probate	probat
	Make	make
(m=1 and not *o)e→	Cease	ceas

Step 5b

(Condition)S1→S2	Word	Stem
(m>1)and*dand*L) →l	petroll	petrol
	Call	call

B. Errors in porter's Algorithm

Error #1:
The conversion from "y" to "i" in the word like Happy→"Happi".

Error #2:
The removal of "ic" or "ical" from words having m=2 and ending with a series of consonant, vowel, consonant, vowel, such as generic, politic.
Political → polit
Generic→ gener

Error #3:
The removal of the suffix "ness" from all words where m=1 and end with consonant, vowel, consonant (cvc) such as witness:
Witness → wit

Error #4:
The suffix "al" is removed from all words where m=2 e.g. admiral, animal.
Admiral →admir

Error #5:
The removal of the suffix "eer" from words with m=2 such as engineer.
Engineer →engin

Error #6
After the removal of "ing" from, removal of one consonant from the word ended by double consonant for some special words having m=1.
Running→ runn
Planning→ plann

C. Solutions for errors in porter's algorithm

These errors are removed by adding new rules of stemming algorithm. Corresponding solution for these errors are follows.
Solution #1:
If the word ends with "y" then do not change it with "i"
Happy→happy
Playing→play

Solution #2:
Usually the words that end by "ic" in step3 or "ical" and having measure of size, m = 2 and consists of a series of consonant, vowel, consonant, vowel, then these are replaced by "e" rather than being removed.
Polite→ polite,
political→ polite

Solution #3:
If the word ends with "ness", m = 1, and ends with consonant, vowel, and a consonant, it is kept as it is.
Witness→ witness
Else it will be removed.

Solution #4:

If it ends by "iral" and m = 2 then it is replaced by "ire". Or if it ends by "al", m = 2, and it consists of a series of consonant, vowel, consonant, vowel, it is replaced by "e".
General→ Genere
Admiral→ admire

Solution #5:
If the word ends with "eer" and having m = 1, then "eer" is replaced by "e".
Engineer→ engine

Solution #6:
If the word ends by nn after removing "ing" and having m=1 then replace "nn" by "n".
Running→ run
Planning→ plan.

These modifications make the enhanced suffix stripping algorithm more efficient in the context of information retrieval system.

III. RESULT ANALYSIS

A. *Analysis on text data*

These modifications make the enhanced algorithm more efficient and more error free .Analysis based on text data is given below.

S.no	No. of words	Meaning full words	Stems
Porter's Algo.	170	120	80
E.S.S.Algo.	170	120	72

Table 1: Analysis on text data

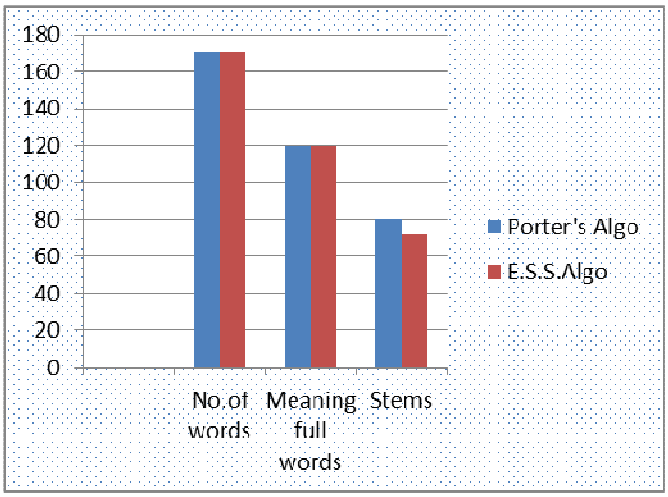

Fig. 2 Analysis on text data

Analysis on text data shows that the porter stemming algorithm reduced the document size upto 47.05% and the enhanced suffix stripping algorithm (E.S.S.Algo.) reduced the document size upto 42.35% so the index size is reduced upto 42.35%. Less index size take less time to map relevant document to the input query so the efficiency of IR system is increased.

B. *Analysis based on errors*

There are mainly two types of errors in stemming process over stemming and under stemming. An ideal stemmer should stem words belonging to the same group to a common stem [7,,8]. If a stemmed group includes more than one unique stem, then the stemmer has made under stemming errors. However, if a stem of a certain group occurs in other stemmed groups, the stemmer has made over stemming errors. This allows the computation of the over stemming and under stemming Indexes (UI and OI) [9 10]. An ideal stemmer should stem words belonging to the same group to a common stem. If a stemmed group includes more than one unique stem, then the stemmer has made under stemming errors [11,12,13]. However, if a stem of a certain group occurs in other stemmed groups, the stemmer has made over stemming errors. This allows the computation of the over stemming and under stemming Indexes (UI and OI) .These are given in table no: 2.

	Word list A		Word list B	
S.no	U.I	O.I	U.I	O.I
E.S.S.Algo.	0.2432	0.0341	0.2127	0.0485
Porter's Algo.	0.3236	0.0462	0.2648	0.0532

Table 2: Error analysis

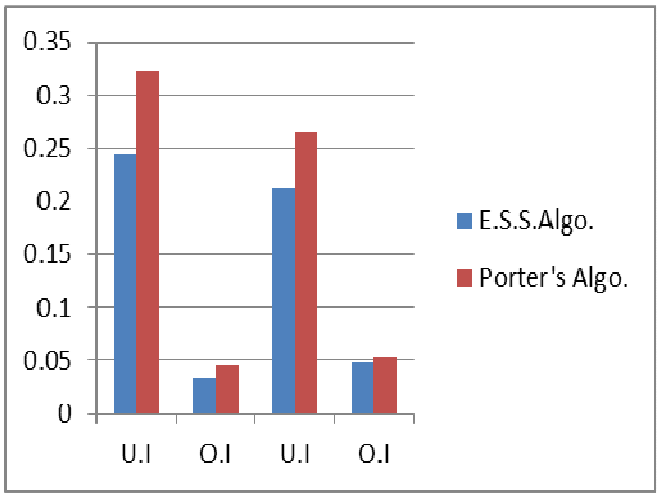

Fig. 3 Error analysis

Above analysis shows that the Enhanced suffix stripping Algorithm have over all low value of under stemming index and over stemming index. By adding new rules the stemming errors are reduced and make more efficient in the context of information retrieval because less error make more relevant document will be retrieved.

IV. CONCLUSION

The Enhanced suffix stripping Algorithm has over all low value of under stemming index and over stemming index. By adding new rules the stemming errors are reduced and make more efficient in the context of information retrieval because less error make more relevant document will be retrieved. A statistical stemmer may be language independent it does not every time give a trustworthy and correct stem. Fig. 2 and 3 shows the better performance of Enhanced suffixed stripping algorithm in the context of index size and number of errors. In above enhanced suffix stripping algorithm index size is reduced upto 42.35% as compared to 47.05% in porter's algorithm. and the over stemming and under stemming index size is also reduced means by using E.S.S Algo in IR system in less amount of time we will get more relevant documents.

V. REFERENCE

[1] Porter M.F. "An algorithm for suffix stripping" Program. **1980**; 14, 130.

[2] Porter M.F. "Snowball: A language for stemming algorithms". **2001**

[3] Eiman Tamah Al-Shammari "Towards An Error-Free Stemming", in Proceedings of ADIS European Conference Data Mining **2008**, pp. 160-163.

[4] "A Survey on various stemming algorithms" International Journal of Computer Engineering In research trends(IJCERT), VOLUME 2, ISSUE 5, May **2015**, PP 310-315

[5] Frakes W.B. "Term conflation for information retrieval". Proceedings of the 7th annual international ACM SIGIR conference on Research and development in information retrieval. **1984**, 383-389.

[6] Frakes William B. "Strength and similarity of affix removal stemming algorithms". ACM SIGIR Forum, Volume 37, No. 1. **2003**, 26-30.

[7] M. Nithya, "Clustering Technique with Porter stemmer and Hyper graph Algorithms for Multi-featured Query Processing", International Journal of Modern Engineering Research (IJMER), Vol.2, Issue.3, pp-960-965, May-June **2012**

[8] Galvez Carmen and Moya-Aneg•n F˜lix. "An Evaluation of conflation accuracy using finite-state transducers". Journal of Documentation 62(3). **2006**, 328-349

[9] J. B. Lovins, "Development of a stemming algorithm," Mechanical Translation and Computer Linguistic., vol.11, no.1/2, pp. 22-31, **1968**.

[10] Harman Donna. "How effective is suffixing?" Journal of the American Society for Information Science. **1991**; 42, 7-15 7.

[11] Funchun Peng, Nawaaz Ahmed, Xin Li and Yumao Lu. "Context sensitive stemming for web search". Proceedings of the 30th annual international ACMSIGIR conference on Research and development in information retrieval. **2007**, 639-646.

[12] R. Sun, C.-H. Ong, and T.-S. Chua. "Mining Dependency Relations for Query Expansion in Passage Retrieval". In SIGIR, **2006**.

[13] Kjetil , Randi, "News Item Extraction for Text Mining in Web Newspapers" WIRI'05,IEEE ,**2009**

Author Profile

Sundar Singh has received B.Tech in Computer Science & Engineering from Gautam Buddh Technical University Lucknow, India in 2012. He is pursuing M.Tech in Advance computing from Maulana Azad National Institute of Technology Bhopal, India. His research area includes Natural Language processing and Information Retrieval.

Dr. R.K. Pateriya is an Associate professor in the Department of Computer Science & Engineering, at Maulana Azad National Institute of Technology, Bhopal, India. He is a member of the IEEE. His current research interests includes Cloud computing, E-commerce, Security, Natural Language Processing and Information Retrieval etc. He has published more than 100 papers in national and international research journals.