ISSN (Online): 2349-7084
GLOBAL IMPACT FACTOR 0.238
ISRA JIF 0.351

INTERNATIONAL JOURNAL OF COMPUTER ENGINEERING IN RESEARCH TRENDS
VOLUME 2, ISSUE 5, May 2015, PP 310-315

A Survey on various Stemming Algorithms

Sundar Singh, R K Pateriya
Computer Science & Engineering Department
Maulana Azad National Institute of Technology
Bhopal, India, 462003
sschaharnitb@gmail.com, pateriyark@gmail.com

Abstract—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. In this paper different stemming algorithms are discussed and compared in terms of usefulness and there limitations.

Index Terms—Stemming, stop word, recall, precision, Text mining, NLP, IR.

1 INTRODUCTION

Stemming is to reduce different grammatical forms of a word like its noun, adjective, verb, and adverb, etc. called parts of speech (POS) to its base form. In other words the goal of stemming is to reduce inflectional forms and sometimes derivationally related forms of a word to a common base form. Stemming process includes several stages e.g. stop word removal, normalization and then stemming. After that indexing and information retrieval (IR) has to be done. Stemming and indexing are very important part of Text Mining, Natural Language Processing (NLP) and Information Retrieval (IR) systems

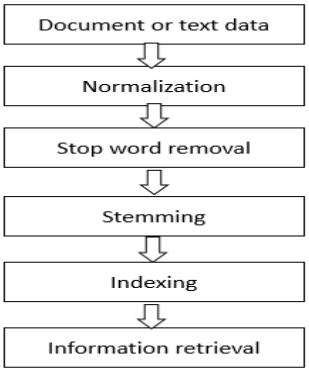

Fig. 1. Various stages in IR

- In the 1st stage either upload a document or type a paragraph or text which is to be stemmed.
- In the 2nd stage 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 stage 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 stage suffix stripping or stemming will be done in this stage all words are converted into their root or base or stem form.
- In the 5th stage indexing of all the stems will be done. Stemming reduced the index size approximately 1/3 of its previous word count.
- In the 6th stage information retrieval will be done.

1.1 Working of stemmer

Stemming and lemmatizing process will do the same work by different methods. Stemming uses a set of rules has to be applied on a word it will not consider the context of sentence and parts of speech of sentence. But Lemmatizing take care of parts of speech and context of the sentence in which the word is used that is to be stemmed. A variety of stemming algorithms have been developed. Each algorithm converts the morphological variants of a word like computer; computing, computed etc. convert to the word 'compute'. Some algorithms may convert them to just 'comput', but that is allowed as long as all the algorithms convert to the same word form is known as the base form. Thus, the key terms of a query or document are indexed by stems rather than by the original words. The main thing is stemming reduces the number of entries of key terms in index of the document or a query which in turn will reduce the processing time of mapping. By converting more number of derived words to their single stem the index size will reduce by which information retrieval processing time will reduce.

2 LICTRATURE REVIEW

2.1 Stemming and Lemmatizing Errors

Stemming and lemmatizing looks like similar. Both the methods reduce a word variant to its 'root'. A very little difference between both the methods is that in stemming the 'root' is obtaining after applying a set of steps containing set of rules but without taking care off the part of speech (POS) or the occurrence of the context of the word. But lemmatizing is the process of alliance many different inflated forms of a word so they can be evaluated as a single item by taking care of verb, gerund, present participle or the POS of the lemmatizing sentence. [8], [9], [10]. The word Morphology means study of forms. Morphological the word have the same base meaning and hence should be converted to the same stem. Words that do not have the same sense should be kept separate either it have similar spelling.

ISSN (Online): 2349-7084

INTERNATIONAL JOURNAL OF COMPUTER ENGINEERING IN RESEARCH TRENDS
VOLUME 2, ISSUE 5, May 2015, PP 310-315

Lemmatization typically examination language and morphology of the words; stemming targeting to remove inflectional forms only and toreturn the root. For example, the word inflations like gone, goes, going will map to the stem 'go'. The word 'went' will not map to the same stem. But the lemmatizing process will convert the word 'went' too to the lemma 'go
Stemming: computing, computation, computerization –>compute and gone, going, goes, go –>go.
Lemmatizing: introduction, introducing, introduces –>introduce and gone, going, goes, went, go –>go.

2.2Stemming Errors

There are two types of errors in stemming

Over-stemming: when the two or more words belong to different root are stemmed to same root is called Over-stemming. This is also known as a false positive [8]. Ex: design, designate->design, etc.

Under-stemming: when the two or more words belong to same root but after stemming converted to different root is called under-stemming. This is also known as a false negative [19].Ex: foot->foot, feet ->feet etc.

2.2 Types of Stemming Algorithms

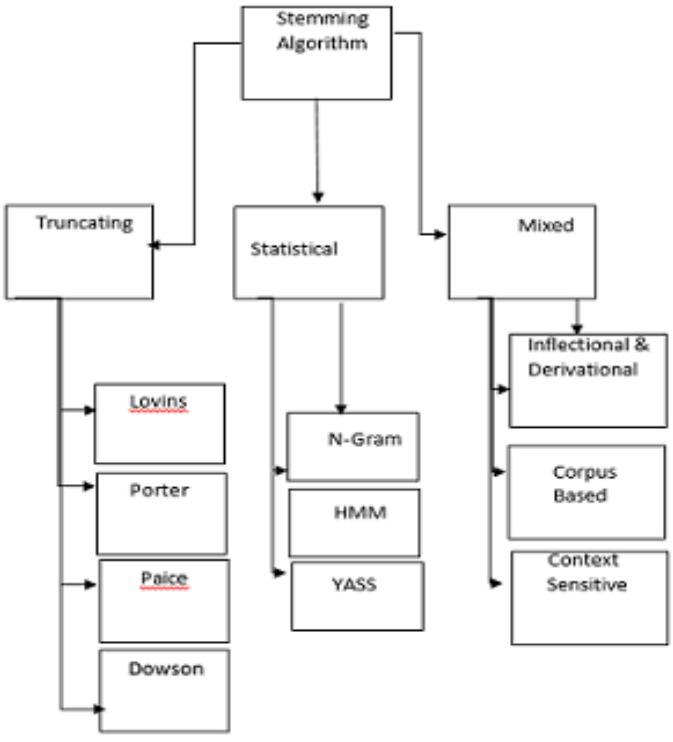

Fig. 2. Various types of stemming algorithms

2.3.1 Truncating Methods

Truncating methods are based on removing the suffix or prefix (commonly known as affixes) of a word. The simpler stripping technique was the Truncate (n) stemmer which truncated a word at the nth symbol i.e. keep n letters and remove rest of the characters. In this method chances of over stemming are increases when the word length is small.

2.2.1.1 Lovins Stemmer

It is the first effective and incredible stemmer proposed by Lovins in 1968. It performs a matching on a table of 294 endings and having 29 conditions with 35 transformation rules, it is based on a longest match method [8]. This stemmer eradicates the longest suffix from a word which is match in the table of ending. After the ending is eradicated, the word is recoded using a different table after that apply various amendments to convert these stems into valid words. It is a single pass algorithm so can remove maximum one suffix from the inflated words.

The pros of this algorithm is due to single pass nature it is very fast and it can easily handle double letters in words like 'running' being malformed to 'run' and also handles many irregular plurals like mouse and mice, index and indices etc.
Cons of the Lovins approach are that it is not much competent and data consuming. Due to many suffixes are not available in the table of endings. The stemming is not performed efficiently.

2.2.1.2 Porters Stemmer

Porters stemming algorithm is one of the most famous stemming methods proposed by martin porter in 1980. It comprises 60 rules in five steps. It is based on suffix Stripping technique. The data is passing through one by one step so it is multi pass algorithm in this each step work like step. 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 [1], [2].

The rule looks like the following:
<Condition><suffix> → <new suffix>
For example, a rule (m>0) EED → EE means "if the word has at least one "vowel and consonant" pare plus EED ending, change the ending to EE". Example "agreed"Converts "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. There are several modification have been done on porter algorithm still it have over stemming and under stemming kind of problems.
Advantages of this algorithm are its efficiency and less time consuming with most widely uses in information retrieval system.
Cons of this algorithm are it has many errors in algorithm and also it has of over stemming and under stemming type of problems.

2.2.1.3 Paice/husk Stemmer

The Paice/Husk Stemmer was produced by Chris Paice at Lancaster University in the late 1980s. It characters a rule execution system and externally stored procedures. This stemmer have iterative algorithm with one table comprising about 120 rules. Indexed by the end letter of a suffix [16]. It tries to find an appropriate rule by the last

ISSN (Online): 2349-7084

INTERNATIONAL JOURNAL OF COMPUTER ENGINEERING IN RESEARCH TRENDS
VOLUME 2, ISSUE 5, May 2015, PP 310-315

character of the word. Each rule stipulates either an exclusion or replacement of an ending. The algorithm was altered to improve the handling of errors in the rules, allow interactive testing. It provide more accurate stems, and add some tractability for carry out finite state automata. Less than 60 lines of code were enhanced or altered without regarding the substitution of the driver. The new driver and compiling decisions make it probable to check the execution of the rules interactively. This is crucial because it is likely for the execution of the rules to get in an endless loop. For example, the rule "y, y, continue" will loop forever when a word ending in "y" is input, the code was modified to prevent infinite looping. If there is no such rule, it terminates. It also terminates if a word starts with a vowel and there are only two letters left or if a word starts with a consonant and there are only three characters left. Otherwise, the rule is applied and the process repeats [16].

The advantage of this algorithm is its flexibility of being able to postulate a new set of rules without broad changes in programming.
The disadvantage of this algorithm is it is a very heavy algorithm and over stemming may occur.

2.2.1.4 Dawson Stemmer

The Dawson stemmer was produced by John Dawson and was first presented in 1974. The stemmer is alike to Lovins as it is a single-pass context-sensitive suffix elimination stemmer. It was established at the literary and linguistics Centre, Cambridge. It is a modification of the lovins approach except that it insurances a considerably more wide-ranging list of about 1200 suffixes. Like lovins it is also a single pass stemmer and therefore is less time consuming. The suffixes are stored in the inverted order indexed by their size and last letter. [9]

The major difference between lovins and Dowson stemmer is the lovins stemmer operates the technique known as storing. This activity is seen as part of the main algorithm and acts a number of conversions based on the letters within the stem but the Dowson stemmer utilizes fractional matching which, efforts to match stems that are alike within certain limits.
The advantage is that it deals with more suffixes than lovins. It is fast in execution with less number of mistakes in stemming.
The disadvantage of this algorithm is it is very difficult and absences in typical reusable implementation.

2.3.2 Statistical Methods

These stemmers are based on statistical analysis and techniques. Most of the methods remove the affixes but after implementing some statistical procedure. Statistical methods are based on counting the number of characters

2.3.2.1 N-Gram Stemmer

This method is language independent. Over here string-similarity approach is used to convert word inflation to its base form. An n-gram is a string of n, adjacent characters extracted from a section of continuous text. To be precise an n-gram is a set of n consecutive characters obtained from a word. The main idea behind this approach is that, similar words will have a high proportion of n-grams in common, get high value of dice coefficient. Higher value of dice's coefficient denotes more similarity. For n equals to 2 or 3, the words extracted are called diagrams or trigrams, respectively [13]. For example, the word 'SUPER' results in the generation of the diagrams.

S, SU, UP, PE, ER, R
And trigrams
S, *SU, SUP, UPE, PER, ER*, R
Where '*' denotes a padding space. There are n+1 such di-gram and n+2 such trigrams in a word containing n characters for example
Computers => co om mp pu ut te er rs.
Computed => co om mp pu ut te ed.

Dice's coefficient (likeness)

Here A and B are the numbers of distinctive di-grams in the first and the second word and C is the distinctive di-grams shared by A and B.
This stemmer has an advantage that it is syntactical independent and hence very useful in many appliances.
The disadvantage is it requires the extra amount of storage for creating and storing the n-grams and indexes and hence is not a very realistic approach.

2.3.2.2HMM Stemmer

It is based on the concept of the Hidden Markov Model (HMMs) to generate statistical stemmer that is finite-state automata where alterations between states are ruled by probability functions. For every transition, the new state emits a symbol with a given probability. This model was

Proposed by Melucci and Orio [14]. This method is not based on supervised learning and does not need a prior linguistic knowledge of the dataset. In this method the probability of each path can be computed and the most probable path is found using the Viterbi coding in the automata graph.
In order to apply HMMs to stemming, a sequence of letters that forms a word can be considered the result of a concatenation of prefix and suffix called affixes. A way to model this process is through an HMM where the states are divided in two disjoint sets initial can be the stems only and the later can be the stems or suffixes. Transitions between states of HMM define as word building process. There are some assumptions that can be considered in this technique.
The advantage of this method is it is unsupervised and hence prior knowledge of the language is not required. It is able to represent the change of purposes power demands through probability distributions.
The disadvantage is it is a little difficult and may over stem the words occasionally. Due to their Markovian nature, they do not take into account the sequence of states leading into any given state.

2.3.2.3 YASS Stemmer

The name YASS is the short form of the Yet another Suffix Striper. This stemmer was proposed by Prasenjit Majumder et.al [18]. The functioning of a stemmer produced by clustering a lexicon without any morphological input is alike to that obtained using standard, rule-based stemmers. Such as Porter's. This stemmer comes under the type of statistical as well as corpus based. It does not depend on syntactical expertise. Retrieval trials by the authors on English, French, and Bengali datasets show that the suggested approach is applicable for languages that are mostly suffixing in nature.

The clusters are formed using hierarchical tactic and distance measures. Then the resulting clusters are considered as equivalence classes and their centroids as the stems. The YASS distance procedures D_1, D_2, D_3 and D_4 are based on a Boolean function pi for penalty. It is defined as

ISSN (Online): 2349-7084

INTERNATIONAL JOURNAL OF COMPUTER ENGINEERING IN RESEARCH TRENDS
VOLUME 2, ISSUE 5, May 2015, PP 310-315

$$p_i = \begin{cases} 0 \text{ if } x_i = y_i & 0 \le i \le \min(n,n') \end{cases} \ldots \quad (1.1)$$

Where X and Y are two strings, X = $x_0x_1x_2 \ldots x_n$ and Y = $y_0y_1y_2 \ldots y_n$. If the strings are of uneven lengths we pad the smaller string with null characters to make the strings lengths equal. Shorter the distance amount shows greater similarity among the strings. The alter distance between two strings of characters is the number of operations needed to renovate one of them into the other.
Advantage of this stemmer can provide strength against distortion constrained. And it arrange for protection against active adversaries.
Disadvantage of this method is its language independency so does not cover whole language.

2.3.3. Mixed Methods

This is another approach to stemming and it involves both the inflectional as well as the derivational morphology consideration. In case ofinflectional the word variants are related to the language specific syntactic variations like plural, gender, case, etc. whereas in derivational the word variants are related to the part-of-speech (POS) of a sentence where the word occurs.

2.3.3.1 Inflectional and Derivational Methods

The linguistics groups at Xerox have developed a number of morphological tools for English which can be used in information retrieval. In particular, they have formed English lexical database which offers a linguistic analysis of any word in the lexicon and recognizes the base form. Xerox linguists have developed a lexical database for English and some other languages also which can analyse and produce inflectional and derivational morphology. The inflectional database diminishes each surface word to the form which can be found in the dictionary [19].

For example, 'government' stems to 'govern' while 'complement' is not reduced to 'comple' since the two forms have different meanings. All stems are effective English terms, and irregular forms are managed correctly. The derivational process uses both suffix and prefix removal, dissimilar most conventional stemming algorithms
The databases are created using finite state transducers, which supports very capable storage and access. This machinery permits the conflation process to act in inverse. It generating all possible surface forms from a single base form. The database starts with a lexicon of about 77 thousand base forms from which it can cause nearly half a million surface forms.
The advantages of this stemmer are that it works well with a large document also and eradicates the prefixes also where ever applicable.
The disadvantage is that the output depends on the lexical database which may not be all-inclusive. Since this method is based on a lexicon, it cannot correctly stem words which are not part of the dictionary.

2.3.3.2 Corpus Stemmer

This technique of stemming was proposed by Xu andCroft in their paper "Corpus-based stemming using co-occurrence of word deviations" [20]. They have recommended a methodology which tries to overcome some of the shortcomings of Porter stemmer. For example, the words policy' and 'police' are conflated though they have a different meaning but the words 'index' and 'indices' are not conflated though they have the same root. Porter stemmer also creates stems which are not real words like 'iteration' becomes 'iter' and 'general' becomes 'gener'. A further difficulty is that although some stemming algorithms may be fit for one corpus, they will produce too many errors on another to define the consequence of word form co-occurrence, the statistical measure is defined as.

$$Em(a,b) = \frac{\quad}{\quad} \ldots \quad (1.2)$$

Where a b are a pair of words, n_a and n_b are the number of occurrences of a and b in the corpus. And nab is the number of times both pare of words fall in a text window of dimensions win in the corpus.
The advantage of this method is it can possibly avoid making conflations that are not appropriate for a given corpus and the outcome is an actual word and not an inadequate stem.
The disadvantage is that you need to develop the statistical measure for every corpus separately and the meting out time increases as in the first step two stemming algorithms are first used before using this method.

2.3.3.3 Context Sensitive Stemmer

In this method stemming unlike the usual method where stemming is done before indexing a document, over here for a Web Search, context sensitive analysis is done using statistical modelling on the query side. This method was proposed by Funchun Peng et al. [21]

Essentially for the words of the input query, the morphological alternates which would be useful for the search are forecast before the query is offered to the search engine. This radically reduces the number of bad expansions, which in turn reduces the amount of additional computation and increases the precision at the same time. After the anticipated word variants from the query have been stemmed, a context sensitive document matching is done for these variants. This conventional strategy serves as a precaution against specious stemming, and it turns out to be very important for improving precision.
The advantage of this stemmer is it increases selective word expansion on the query side and conservative word occurrence matching on the document side.
The disadvantage is the processing time and the complicated nature of the stemmer. There can be errors in finding the noun phrases in the query and the closeness of words.

3 STEMMER STRENGTH MEASURING TERMS

3.1 Recall

Recall is defined as the fraction of relevant retrieved document and the entire relevant document present on the server [11].

$$\text{Recall} = \frac{\quad}{\quad} \ldots \quad (1.3)$$

3.2 Precision

Precision is defined as the fraction of relevant retrieved document and the entire retrieved document from the server.

$$\text{Precision} = \frac{\quad}{\quad} \ldots \quad (1.4)$$

3.3 F-measure

F-measure is defined as the harmonic mean of recall and precision.

$$\text{F-measure} = \frac{\quad}{\quad} \ldots \quad (1.5)$$

Where R denote recall and P denote precision for making an efficient information retrieval system recall and precision should be high so F-measure should be high [11].

3.4 Mean word conflation

This metric is clearly hooked on the number of words processed, but for a word collection of given size, a higher value indicates a heavyweight stemmer. The value is easily calculated as follows:[5]

$$MCW = \frac{\cdot\cdot}{\cdot}\ldots \qquad (1.6)$$

Where MWC is Mean number of words per conflation class and N is Number of unique words before Stemming and S is Number of unique stems after Stemming[19]. Higher value of MWC denote better strength.

3.5 Index Compression Factor

The Index Compression Factor represents the extent to which a pool of unique words is reduced (compressed) by stemming. The idea being that the heavier the Stemmer, the greater the Index Compression Factor. This can be calculatedas

$$ICF = \frac{\cdot\cdot-\cdot}{\cdot\cdot}\ldots \qquad (1.7)$$

Where IFC is index compression factor N is Number of distinctive words before Stemming and S is Number of unique stems after Stemming higher value of IFC denote heigher strength of stemming algorithm.

4 Conclusion

Fig. 1 shows the steps used in stemming and IR.As can be realized from all the algorithms that have been discussed so far, there is a lot of likeness between the stemming algorithms and if one algorithm scores better in one part, the other does superior in some other part. In fact, none of them give 100% productivity but are satisfactory useful to the text mining, NLP or IR applications. The main variance lies in using either a rule-based approach or a linguistic one. A rule based tactic may not always give correct output and the stems produced may not at all times be correct words. Insofar as the linguistic approach is concerned, since these approaches are based on a lexicon, words external to the lexicon are not stemmed properly. It is of highest importance that the lexicon being used is totally intensive, which is concern of language study. A statistical stemmer may be language independent it does not every time give a trustworthy and correct stem. The problem of over stemming and under stemming can be reduced only if thecomposition as well as the POS of the sentence is taken into consideration. Combination with a dictionary look-up can support in decreasing the errors and converting stems to words. However no perfect stemmer has been designed till now to match all the requirements.

5 Future work

A lot of research work has been done in developing stemmers. There stillremains a lot to be done to improve recall as well as precision. There are many faults are present in current stemming algorithm we have to reduce such errors by modifying available algorithms or making new algorithm. Also there should be a need to reduce over stemming and under stemming.A stemmer that uses the syntactical as well as the semantically information to reduce stemming errors should be developed.

6 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] 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.
[5] Frakes William B. “Strength and similarity of affix removal stemming algorithms”. ACM SIGIR Forum, Volume 37, No. 1. 2003, 26-30.
[6] 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
[7] 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
[8] J. B. Lovins, “Development of a stemming algorithm,” Mechanical Translation and Computer Linguistic., vol.11, no.1/2, pp. 22-31, 1968.
[9] Harman Donna. “How effective is suffixing?” Journal of the American Society for Information Science. 1991; 42, 7-15 7.
[10] Kjetil, Randi, “News Item Extraction for Text Mining in Web Newspapers” WIRI’05, IEEE, 2009
[11] Kraaij Wessel and Pohlmann Renee. “Viewing stemming as recall enhancement”. Proceedings of the 19thannual international ACM SIGIR conference on Research and development in information retrieval. 1996, 40-48.
[12] Krovetz Robert. “Viewing morphology as an inference process”. Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval. 1993, 191-202.
[13] Mayfield James and McNamee Paul. “Single N-gram stemming”. Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval. 2003, 415-416.
[14] Melucci Massimo and Orio Nicola. “A novel method for stemmer generation based on hidden Markov models”. Proceedings of the twelfth international conference on Information and knowledge management. 2003, 131-138.
[15] Mladenic Dunja. “Automatic word lemmatization”. Proceedings B of the 5th International Multi-Conference Information Society IS. 2002, 153-159. [14] Paice Chris D. “Another stemmer”. ACM SIGIR Forum, Volume 24, No. 3. 1990, 56-61.
[16] Paice Chris D. “An evaluation method for stemming algorithms”. Proceedings of the 17th annual international ACM SIGIR conferenceon Research and development in information retrieval. 1994, pp. 42-50.
[17] Plisson Joel, Lavrac Nada and Mladenic Dunja. “A rule based approach to word lemmatization”. Proceedings C of the 7th International Multi-Conference Information Society IS. 2004
[18] Prasenjit Majumder, Mandar Mitra, Swapan K. Parui, Gobinda Kole, Pabitra Mitra and Kalyankumar Datta. “YASS: Yet another suffix stripper”. ACM Transactions on Information Systems. Volume 25, Issue 4. 2007, Article No. 18.

[19] Hull David A. and Grefenstette Gregory. “A detailed analysis of English stemming algorithms”. Rank Xerox Research Center Technical Report. 1996.

[20] Xu Jinxi and Croft Bruce W. “Corpus-based stemming using co-occurrence of word variants”.ACM Transactions on Information Systems. Volume 16, Issue 1. 1998, 61-81.

[21] 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.

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

[23] Toman Michal, Tesar Roman and Jezek Karel. “Influence of word normalization on text classification”. The 1st International Conference on Multidisciplinary Information Sciences & Technologies. 2006, 354-358.

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, atMaulana Azad National Institute of Technology, Bhopal, India. He is amember 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