International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868
Foundation of Computer Science FCS, New York, USA
Volume 4– No.3, September 2012 – www.ijais.org

Stemming Algorithms: A Comparative Study and their Analysis

Deepika Sharma [ME CSE]
Department of Computer Science and Engineering, Thapar University
Patiala, Punjab, India

ABSTRACT
Stemming is an approach used to reduce a word to its stem or root form and is used widely in information retrieval tasks to increase the recall rate and give us most relevant results. There are number of ways to perform stemming ranging from manual to automatic methods, from language specific to language independent each having its own advantage over the other. This paper represents a comparative study of various available stemming alternatives widely used to enhance the effectiveness and efficiency of information retrieval**.**

Keywords
Information Retrieval, Stemming Algorithm, Conflation Methods

1. INTRODUCTION
With the enormous amount of data available online, it is very essential to retrieve accurate data for some user query. There are lots of approaches used to increase the effectiveness of online data retrieval. The traditional approach used to retrieve data for some user query is to search the documents present in the corpus word by word for the given query. This approach is very time consuming and it may miss some of the related documents of equal importance. Thus to avoid these situations, Stemming has been extensively used in various Information Retrieval Systems to increase the retrieval accuracy.

Stemming is the conflation of the variant forms of a word into a single representation, i.e. the stem. For example, the terms presentation, presenting, and presented could all be stemmed to present. The stem does not need to be a valid word, but it must capture the meaning of the word. In Information Retrieval Systems stemming is used to conflate a word to its various forms to avoid mismatches between the query being asked by the user and the words present in the documents. For example if a user wants to search for a document on "How to cook" and submits a query on "cooking" he may not get all the relevant results. However, if the query is stemmed, so that "cooking" becomes "cook", then retrieval will be successful.

Stemming has been extensively used to increase the performance of Information Retrieval Systems. For some International languages like Hebrew, Portuguese, Hungarian [3], Czech, and French and for many Indian languages like Bengali, Marathi, and Hindi [2] stemming increase the number of documents retrieved by between 10 and 50 times. For English though the results are less dramatic but better then the baseline approach where no stemming is used. Stemming is also used to reduce the size of index files. Since a single stem typically corresponds to several full terms, by storing stems instead of terms, compression factor of 50 percent can be achieved.

2. CONFLATION METHODS
For achieving stemming we need to conflate a word to its various variants. Figure 1 shows a various conflation methods that can be used in stemming. Conflation of words or so called stemming can either be done manually by using some kind of regular expressions or automatically using stemmers. There are four automatic approaches namely Affix Removal Method, Successor Variety Method, n-gram Method and Table lookup method [1] [7].

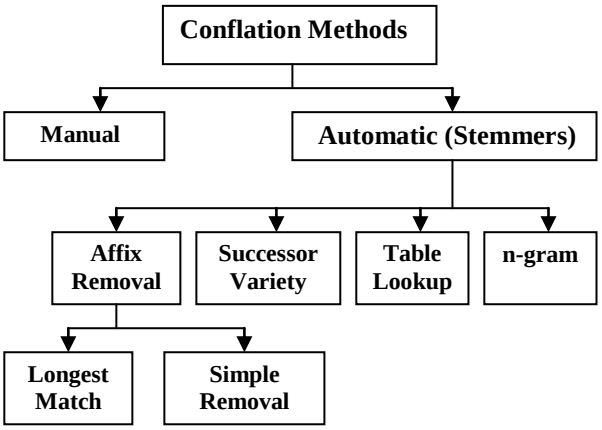

Figure 1 Conflation Method

2.1 Affix Removal Method
The affix removal method removes suffix or prefix from the words so as to convert them into a common stem form. Most of the stemmers that are currently used use this type of approach for conflation. Affix removal method is based on two principles one is iterations and the other is longest match.
An iterative stemming algorithm is simply a recursive procedure, as its name implies, which removes strings in each order-class one at a time, starting at the end of a word and working toward its beginning. No more than one match is allowed within a single order-class, by definition. Iteration is

usually based on the fact that suffixes are attached to stems in a "certain order, that is, there exist order-classes of suffixes.
The longest-match principle states that within any given class of endings, if more than one ending provides a match, the one which is longest should be removed. The first stemmer based on this approach is the one developed by Lovins (1968); MF Porter (1980) also used this method. However, Porter's stemmer is more compact and easy to use then Lovins. YASS is another stemmer based on the same approach; it is however language independent is nature.

2.2 Successor Variety Method

Successor variety stemmers [8] use the frequencies of letter sequences in a body of text as the basis of stemming. In less formal terms, the successor variety of a string is the number of different characters that follow it in words in some body of text. Consider a body of text consisting of the following words, for example.

back, beach, body, backward, boy

To determine the successor varieties for "battle," for example, the following process would be used. The first letter of battle is "b." "b" is followed in the text body by four characters: "a," "e," and "o." Thus, the successor variety of "b" is three. The next successor variety for battle would be one, since only "c" follows "ba" in the text When this process is carried out using a large body of text, the successor variety of substrings of a term will decrease as more characters are added until a segment boundary is reached. At this point, the successor variety will sharply increase. This information is used to identify stems.

2.3 Table Lookup method

Terms and their corresponding stems can also be stored in a table. Stemming is then done via lookups in the table. One way to do stemming is to store a table of all index terms and their stems. Terms from queries and indexes could then be stemmed via table lookup [6]. Using B-tree or Hash table, such lookups would be very fast. For example, presented, presentable, presenting all can be stemmed to a common stem present.
There are problems with this approach. The first is that there for making these lookup tables we need to extensively work on a language. There will be some probability that these tables may miss out some exceptional cases. Another problem is the storage overhead for such a table.

2.4 n- gram Method

Another method of conflating terms called the shared digram method given in 1974 by Adamson and Boreham [9]. A digram is a pair of consecutive letters. Besides digrams we can also use trigrams and hence it is called n-gram method in general [4]. In this approach, pairs of words are associated on the basis of unique digrams they both possess. For calculating this association measures we use Dice's coefficient [1]. For example, the terms information and informative can be broken into digrams as follows.

information => in nf fo or rm ma at ti io on
unique digrams = in nf fo or rm ma at ti io on
informative => in nf fo or rm ma at ti iv ve
unique digrams = in nf fo or rm ma at ti iv ve

Thus, "information" has ten digrams, of which all are unique, and "informative" also has ten digrams, of which all are unique. The two words share eight unique digrams: in, nf, fo, or, rm, ma, at, and ti.

Once the unique digrams for the word pair have been identified and counted, a similarity measure based on them is computed. The similarity measure used is Dice's coefficient, which is defined as: $S = \frac{2C}{A + B}$

where *A* is the number of unique digrams in the first word, *B* the number of unique digrams in the second, and *C* the number of unique digrams shared by *A* and *B*. For the example above, Dice's coefficient would equal (2 x 8) / (10 + 10) = .80. Such similarity measures are determined for all pairs of terms in the database. Once such similarity is computed for all the word pairs they are clustered as groups. The value of Dice coefficient gives us the hint that the stem for these pair of words lies in the first unique 8 digrams.

3. CLASSIFICATION OF STEMMING ALGORITHM

Stemming algorithms can be broadly classified into two categories, namely Rule – Based and Statistical.

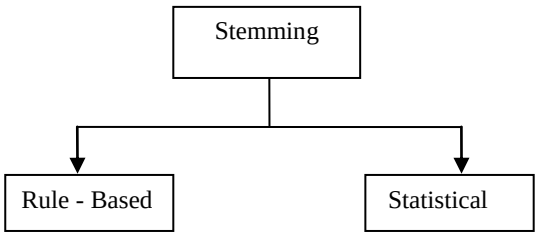

Figure 2 Types of Stemming Approach

Rule based Stemmer encodes language specific rules where as statistical stemmer employs statistical information from a large corpus of a given language to learn the morphology.

3.1 Rule Based Approach

In a rule based approach language specific rules are encoded and based on these rules stemming is performed. In this approach various conditions are specified for converting a word to its derivational stem, a list of all valid stems are given and also there are some exceptional rules which are used to handle the exceptional cases. In Lovins stemmer, stemming comprises of two phases [11]: In the first phase, the stemming algorithm retrieves the stem from a word by removing its longest possible ending by matching these endings with the list of suffixes stored in the computer and in the second phase spelling exceptions are handled. For example the word "absorption" is derived from the stem "absorpt" and "absorbing" is derived from the stem "absorb". The problem of the spelling exceptions arises in the above case when we try to match the two words "absorpt" and "absorb". Such exceptions are handled very carefully by introducing recording and partial matching techniques in the stemmer as post stemming procedures.

Recording [11] occurs immediately following the removal of an ending and makes such changes at the end of the resultant stem as are necessary to allow the ultimate matching of varying stems. These changes may involve turning one stem into another (e.g. the rule rpt → rb changes absorpt to absorb), or changing both stems involved by either recording their terminal consonants to some neutral element (absorb → absor ∂, absorpt → absor∂), or removing some of these letters entirely, that is, changing them to nullity (absorb → absor, absorpt → absor).

The main difference between recording and partial matching is that a recording procedure is a part of stemming algorithm whereas partial matching procedure is applied on the output of stemming algorithm where the stems derived from the catalogue terms are being searched for matches to the user's query.

Apart form Lovins method; one more rule based method is given by MF Porter which comprises of a set of conditional rules [10]. These conditions are either applied on the stem or on the suffix or on the stated rules. As per the conditions, a word can be represented in a general form like:

$$[C](VC)^m[V]$$

Where C represents a list of consonants, V represents a list of vowels and m represents the measure of any word. For example:

m=0 RA, EE, BI, AT

m=1 TREES, OATS, RATES

m=2 TEACHER, TROUBLES, SITUATION

The general rule for removing a suffix is given as:

$$(\text{condition})S1 \rightarrow S2$$

Where, condition represents a stem and if the condition is satisfied then suffixes S1 is replaced by suffix S2. For example

$$(m>1)\text{ION} \rightarrow$$

Here S1 is ION and S2 is null. This would map EDUCATION to EDUCAT, since EDUCAT is a word part for which m=2.

3.1.1 ADVANTAGES

1. Rule Based stemmers are fast in nature i.e. the computation time used to find a stem is lesser.
2. The retrieval results for English by using Rule Based Stemmer are very high.

3.1.2 DISADVANTAGES

1. One of the main disadvantages of Rule Based Stemmer is that one need to have extensive language expertise to make them.
2. The procedure used in this approach handles individual words: it has no access to information about their grammatical and semantic relations with one another.
3. The amount of storage required to store rules for stem extraction from the words and also to store the exceptional cases.
4. These stemmers may apply over stemming and under stemming to the words.

3.2 Statistical Approach

Statistical stemming is an effective and popular approach in information retrieval [16] [5]. Some recent studies [17] [18] show that statistical stemmers are good alternatives to rule-based stemmers. Additionally, their advantage lies in the fact that they do not require language expertise. Rather they employ statistical information from a large corpus of a given language to learn morphology of words. Lot of research has been done in the area of statistical stemming method, some of the latest works are stated below:

3.2.1 YET ANOTHER SUFFIX STRIPPER (YASS)

Most popular stemmers encode a large number of languages specific rules built over a length of time. Such stemmers with comprehensive rules are available only for a few languages. In the absence of extensive linguistic resources for certain languages, statistical language processing methods have been successfully used to improve the performance of IR systems. Yet another suffix stripper (YASS) is one such statistics based language independent stemmer [18]. Its performance is comparable to that of Porter's and Lovin's stemmers, both in terms of average precision and the total number of relevant documents retrieved the challenge of retrieval from languages with poor resources.

In this approach, a set of string distance measures [12] is defined, and complete linkage clustering is used to discover equivalence classes from the lexicon. The string distance measure is used to check the similarity between two words by calculating the distance between two strings , the distance function maps a pair of string a and b to a real number r, where a smaller value of r indicates greater similarity between a and b. A set of string distance measures $\{D_1, D_2, D_3, \text{and } D_4\}$ for clustering the words. The main reason to calculate these distances is to find long matching prefixes and to penalize an early mismatch.

Given two strings $X = x_0x_1.....x_n$ and $Y = y_0y_1.....y_{n'}$ we first define a Boolean function p_i as penalty for an early mismatch:

$$p_i = \begin{cases} 0 \text{ if } x_i = y_i \quad 0 \le i \le \min(n,n') \\ 1 \text{ otherwise} \end{cases}$$

Thus, p_i is 1 if there is a mismatch in the ith position of X and Y. If X and Y are of unequal length, we pad the shorter string with null characters to make the string lengths equal. Let the length of the string be n+1. We define D_1 as follows:

$$D_1(X, Y) = \sum_{i=0}^{n} \frac{1}{2^i} p_i \qquad (1)$$

Accordingly we define D_2, D_3 and D_4 as follows:

$$D_2(X, Y) = \frac{1}{m} \times \sum_{i-m}^{n} \frac{1}{2^{i-m}} \text{ if } m > 0, \infty \text{ otherwise} \quad (2)$$

$$D_3(X, Y) = \frac{n - m + 1}{m} \times \sum_{i-m}^{n} \frac{1}{2^{i-m}} \text{ if } m > 0, \infty \text{ otherwise} \quad (3)$$

$$D_4(X, Y) = \frac{n - m + 1}{n + 1} \times \sum_{i-m}^{n} \frac{1}{2^{i-m}} \quad (4)$$

Where, m represents the position of first mismatch between X and Y. In figure 3, we consider two pair of strings {independence, independently} and {indecent, independence} and value of various distance measure for these two pair of words is calculated as below. Clearly we can infer that indecent and independent are farther apart from independence and independently.

0	1	2	3	4	5	6	7	8	9	10	11	12
I	N	D	E	P	E	N	D	E	N	C	E	*
I	N	D	E	P	E	N	D	E	N	T	L	Y

$$D_1 = \frac{1}{2^{11}} + \frac{1}{2^{12}} = 0.00073$$

$$D_2 = \frac{1}{11} \times \left(\frac{1}{2^0} + \frac{1}{2^1} \right) = 0.1363$$

$$D_3 = \frac{2}{11} \times \left(\frac{1}{2^0} + \frac{1}{2^1} \right) = 0.2727$$

$$D_4 = \frac{2}{13} \times \left(\frac{1}{2^0} + \frac{1}{2^1} \right) = 0.2307$$

Edit Distance = 2

0	1	2	3	4	5	6	7	8	9	10	11
I	N	D	E	C	E	N	T	*	*	*	*
I	N	D	E	P	E	N	D	E	N	C	E

$$D_1 = \frac{1}{2^4} + \frac{1}{2^5} + \ldots\ldots + \frac{1}{2^{11}} = 0.1245$$

$$D_2 = \frac{1}{4} \times \left(\frac{1}{2^0} + \frac{1}{2^1} + \ldots\ldots + \frac{1}{2^{11-4}} \right) = 0.4980$$

$$D_3 = \frac{8}{4} \times \left(\frac{1}{2^0} + \frac{1}{2^1} + \ldots\ldots + \frac{1}{2^{13-11}} \right) = 3.984$$

$$D_4 = \frac{8}{12} \times \left(\frac{1}{2^0} + \frac{1}{2^1} + \ldots\ldots + \frac{1}{2^{13-11}} \right) = 1.328$$

Edit Distance = 8

Figure 3 Calculations of Various Distance Measures

This distance counts the minimum number of edit operations (inserting, deleting, or substituting a letter) required to transform one string to the other. Once similarity between pair of words have been calculated using distance measure, cluster of the words are made by using complete linkage algorithm. In the complete-linkage algorithm [13], the similarity of two clusters is calculated as the minimum similarity between any member of one cluster and any member of the other, the probability of an element merging with a cluster is determined by a least similar member of the cluster.

3.2.2 GRAPH BASED STEMMER (GRAS)

GRAS is a graph based language independent stemming algorithm for information retrieval [19]. The following features make this algorithm attractive and useful: (1) retrieval effectiveness, (2) generality, that is, its language-independent nature, and (3) low computational cost. The steps that are followed in this approach can be summarized as below:

1. Find long common prefix among the word pairs present in the documents. For this, we consider the word-pairs of the form $W_1 = PS_1 \& W_2 = PS_2$ where, P is the long common prefix between $W_1 \& W_2$.
2. The suffix pair S1 & S2 should be valid suffixes i.e. if other word pairs also have a common initial part followed by these suffixes such that $W'_1 = P'S_1 \& W'_2 = P'S_2$. Then, $S_1 \& S_2$ is the pair of candidate suffix if large number of word pairs is of this form. Thus, suffixes are considered in pair rather than individually.
3. Look for pairs that are morphological related i.e. if
   -They share a non-empty common prefix.
   -The suffix pair is a valid candidate suffix pair.
4. These words relationships will be modelled using a Graph where nodes represent the words and edges are used to connect the related words.
5. Pivot node is identified i.e. pivot is considered that node which is connected by edges to a large number of other nodes.
6. In the final step, a word that is connected to a pivot is put in the same class as the pivot if it shares many common neighbours with the pivot.

Once such words classes are formed, stemming is done by mapping all the words in a class to the pivot for that class. This stemming algorithm has outperformed Rule-Based Stemmer, Statistical Stemmer (YASS, Linguistica [15] etc), and Baseline Strategy.

3.2.3 ADVANTAGES

1. Statistical stemmers are useful for languages having scarce resources. Like the Asian languages are heavily used in Asian Sub Continent but very less research is done on these languages.
2. This approach yields best retrieval results for suffixing languages or the languages which are morphologically more complex like French, Portuguese, Hindi, Marathi, and Bengali rather than English.
3. They are considered as Recall – Enhancing Devices as they increase the value of recall at a given rate.

3.2.4 DISADVANTAGES

1. Most of the statistical stemmer does their statistical analysis based on some sample of the actual corpus. As sample size decreases, the possibility of covering most morphological variants will also decrease. Naturally, this would result in a stemmer with poorer coverage.

2. For the Bengali lexicon, there are few instances where two semantically different terms fall in the same cluster due to their string similarity. For example, *Akram* (the name of a cricketer from Pakistan) and *akraman* (to attack) fall in the same cluster, as they share a significant prefix [18]. Such cases might lead to unsatisfactory results.

3. Statistical Stemmers are time consuming because for these stemmers to work we need to have complete language coverage, in terms of morphology of words, their variants etc.

4. COMPARISION AMONG THESE APPROACHES

Here we will compare the performance of various stemming approaches discussed till now. In this comparison we consider one rule-based approach and compare it with statistical approaches like YASS and GRAS. The parameters used in this comparison are each stemmer's strength and the computation time required by each stemmer to compute the stem.

4.1 Stemmer Strength

We now present a comparative study of various stemmers in terms of the stemmer strength. Stemmer Strength [14] generally represents the extent to which a stemming method changes words to its stems. One well-known measure of stemmer strength is the average number of words per conflation class. Formally, if Na, Nw, and Ns denote the mean number of words per conflation class, the number of distinct words before stemming and the number of unique stems after stemming respectively, then $Na = \frac{N_w}{N_s}$ [19].

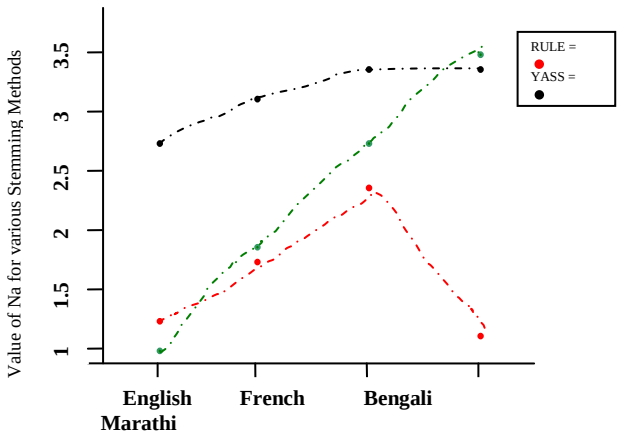

Figure 4 Stemmer Strength

Figure 4 gives the value of N_a for various stemming methods, clearly a higher value of N_a indicates a more aggressive stemmer. Among the three stemmers that we have considered YASS appears to be particularly aggressive on all languages and produces largest N_a value for English, French and Bengali. On the other hand, GRAS is the most aggressive on Marathi while it works equally well as rule- based stemmer for other languages like English, French and Bengali.

4.2 Computation Time

The comparison above clearly shows that YASS outperforms all other stemmer. One more parameter that is used by researchers for comparing the performance of stemmers is computation time which includes the time from submitting a query to its processing and final retrieval. Figure 5 clearly shows that for equal number of words in various languages like English, French, Bengali and Marathi the computation time of YASS is far more than its closest competitor GRAS [19]. So, we concluded that GRAS is far faster than YASS. In GRAS, two aspects that influence the processing time are the density of graph, that is, average degree of a node, and the length of the suffix.

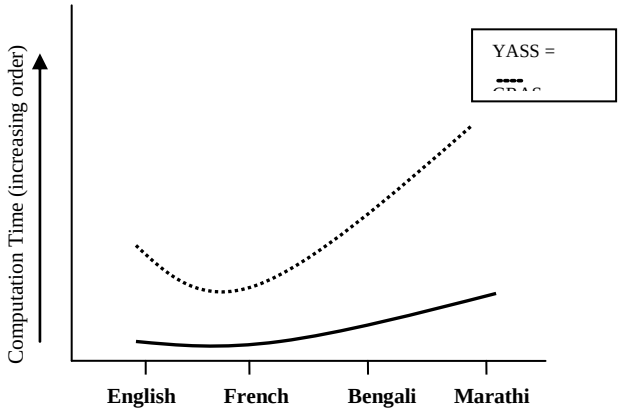

Figure 5 Computation Time

5. CONCLUSION

In the past few years, the amount of information on the Web has grown exponentially. The information present on the Web is practically on all topics and in various languages. Some of these languages have not received much attention and for which these language resources are scarce. To make this available information useful, it has to be indexed and made searchable by an Information Retrieval System. Stemming is one such approach used in indexing process

We have presented a comparative study of various stemming methods. In this we studied that stemming significantly increases the retrieval results for both rule based and statistical approach. It is also useful in reducing the size of index files as the number of words to be indexed are reduced to common forms or so called stems. The performance of statistical stemmers is far superior to some well-known rule-based stemmers and among statistical based stemmers GRAS has outperformed YASS which is a clustering based suffix stripping algorithm. But the main drawback that we have seen in these statistical stemmers is the poor coverage of language i.e. they do not include all the documents in the corpus to make the statistical analysis as it is very time consuming rather they considers sample of documents from the corpus for this analysis and this small collection may lead to poor coverage of

the words. The performance of GRAS is also dependent on the density of the graph but studies have shown that it is capable of handling an interesting class of languages and improves performance of Mono-lingual information retrieval significantly with a low computation cost and in comparatively low processing time.

6. FUTURE SCOPE

Despite of the fact that stemming greatly enhances the performance of Information Retrieval Systems there are still some open issues in this field that are to be dealt properly. In GRAS most of the time is spent on graph construction. These graphs are dynamic in nature as more words are introduced in the corpus, more nodes will be made and graph will become more complex and dense. Also the size of the sample that is considered in statistical stemming is under debate, if smaller size of the sample is considered then stemming will be faster but language coverage will be in doubt and if larger samples are taken then stemming itself will take very long time. So, some optimum sample must be considered that covers maximum lexicon of a language.

7. REFERENCES

[1] WB Frakes, 1992,"Stemming Algorithm ", in "Information Retrieval Data Structures and Algorithm", Chapter 8, page 132-139.

[2] A. Ramanathan and D. Rao, 2003." A lightweight stemmer for Hindi". In Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL), on Computational Linguistics for South Asian Languages (Budapest, Apr.) Workshop.

[3] J. Savoy 2008." Searching strategies for the Hungarian language". Inf. Process. Manage. 44, 1, 310–324.

[4] P. McNamee, and J. Mayfield 2004." Character n-gram tokenization for European language text retrieval", Inf. Retr. 7(1-2), 73–97.

[5] D.W. Oard, G.A. Levow and C.I. Cabezas 2001. CLEF experiments at Maryland:" Statistical stemming and back off translation". In Revised Papers from the Workshop of Cross-Language Evaluation Forum on Cross-Language Information Retrieval and Evaluation (CLEF), Springer, London, 176–187.

[6] WB Frakes 1984. "Term Conflation for Information Retrieval" in Research and Development in Information Retrieval, ed. C. van Rijsbergen. New York: Cambridge University Press.

[7] WB Frakes 1992 "LATTIS: A Corporate Library and Information System for the UNIX Environment," Proceedings of the National Online Meeting, Medford, N.J.: Learned Information Inc., 137-42.

[8] M. Hafer and S. Weiss 1974. "Word Segmentation by Letter Successor Varieties," Information Storage and Retrieval, 10, 371-85.

[9] G. Adamson and J. Boreham 1974. "The Use of an Association Measure Based on Character Structure to Identify Semantically Related Pairs of Words and Document Titles," Information Storage and Retrieval, 10, 253-60.

[10] M. F. Porter 1980. "An Algorithm for Suffix Stripping Program", 14(3), 130-37.

[11] J. B. Lovins 1968. "Development of a Stemming Algorithm." Mechanical Translation and Computational Linguistics, 11(1-2), 22-31.

[12] V. I. Levenstein 1966. Binary codes capable of correcting deletions, insertions and reversals. Commun. ACM 27, 4, 358–368

[13] A. K. Jain, M.N. Murthy, and P. J. Flynn 1999. "Data clustering": A review. ACM Comput. Surv. 31, 3, 264–323.

[14] WB Frakes and C. J. Fox 2003. Strength and similarity of affix removal stemming algorithms. SIGIR.

[15] J. Goldsmith 2001." Linguistica: Unsupervised learning of the morphology of a natural language". Comput. Linguist. 27, 2, 153–198.

[16] J. Xu and W. B. Croft 1998." Corpus-based stemming using co occurrence of word variants". ACM Trans. Inf. Syst. 16, 1, 61–81.

[17] M. Bacchin, N. Ferro, and M. Melucci 2005. "A probabilistic model for stemmer generation". Inf. Process. Manage. 41, 1, 121–137.

[18] P. Majumder, M Mitra, S.K. Parui, and G. Kole (ISI), P. Mitra (IIT), and K.K. Dutta. "YASS: Yet another Suffix Stripper", published in ACM Transaction on Information System (TOIS), Volume 25 Issue 4, October 2007, Chapter 18, Page 5-6.

[19] JH Paik, Mandar Mitra, Swapan K. Parui, Kalervo Jarvelin, "GRAS : An effective and efficient stemming algorithm for information retrieval", published in ACM Transaction on Information System (TOIS), Volume 29 Issue 4, December 2011, Chapter 19, page 20-24

.