UNLV | University Libraries
University of Nevada, Las Vegas

UNLV Theses, Dissertations, Professional Papers, and Capstones

12-2010

Study of stemming algorithms

Savitha Kodimala
University of Nevada, Las Vegas

Follow this and additional works at: https://digitalscholarship.unlv.edu/thesesdissertations

 Part of the Computational Linguistics Commons, Computer Engineering Commons, and the Library and Information Science Commons

Repository Citation

Kodimala, Savitha, "Study of stemming algorithms" (2010). *UNLV Theses, Dissertations, Professional Papers, and Capstones*. 754.
https://digitalscholarship.unlv.edu/thesesdissertations/754

This Thesis is protected by copyright and/or related rights. It has been brought to you by Digital Scholarship@UNLV with permission from the rights-holder(s). You are free to use this Thesis in any way that is permitted by the copyright and related rights legislation that applies to your use. For other uses you need to obtain permission from the rights-holder(s) directly, unless additional rights are indicated by a Creative Commons license in the record and/or on the work itself.

This Thesis has been accepted for inclusion in UNLV Theses, Dissertations, Professional Papers, and Capstones by an authorized administrator of Digital Scholarship@UNLV. For more information, please contact digitalscholarship@unlv.edu.

www.manaraa.com

STUDY OF STEMMING ALGORITHMS

By

Savitha Kodimala

Bachelor of Technology, Computer Science
Jawaharlal Nehru Technological University, India
2008

A thesis submitted in partial fulfillment
of the requirements for the

Master of Science Degree in Computer Science
School of Computer Science
Howard R. Hughes College of Engineering

Graduate College
University of Nevada, Las Vegas
December 2010

www.manaraa.com

UNLV
UNIVERSITY OF NEVADA LAS VEGAS

THE GRADUATE COLLEGE

We recommend the thesis prepared under our supervision by

Savitha Kodimala

entitled

Study of Stemming Algorithms

be accepted in partial fulfillment of the requirements for the degree of

Master of Science in Computer Science
School of Computer Science

Dr.Kazem Taghva, Committee Chair

Dr.Ajoy K.Datta, Committee Member

Dr.Laxmi Gewali, Committee Member

Dr.Venkatesan Mutukumar, Graduate Faculty Representative

Ronald Smith, Ph. D., Vice President for Research and Graduate Studies
and Dean of the Graduate College

August 2010

www.manaraa.com

ABSTRACT

Study of Stemming Algorithms

by

Savitha Kodimala

Dr. Kazem Taghva, Examination Committee Chair
Professor of Computer Science
University of Nevada, Las Vegas

Automated stemming is the process of reducing words to their roots. The stemmed words are typically used to overcome the mismatch problems associated with text searching.

In this thesis, we report on the various methods developed for stemming. In particular, we show the effectiveness of n-gram stemming methods on a collection of documents.

www.manaraa.com

TABLE OF CONTENTS

ABSTRACT iii

LIST OF FIGURES v

ACKNOWLEDGEMENTS vi

CHAPTER 1 INTRODUCTION 1
1.1 Thesis Structure 2

CHAPTER 2 INFORMATION RETRIEVAL 3
2.1 Boolean Retrieval 4
2.2 Co-ordinate Matching 7
2.3 Vector Space Model 11
2.4 Probabilistic Retrieval Model 12
2.5 Language Model 13

CHAPTER 3 STEMMING ALGORITHMS 16
3.1 Types of Stemming Algorithms 18
3.1.1 Table Look Up Approach 18
3.1.2 Successor Variety 18
3.1.3 N-Gram stemmers 20
3.1.4 Affix Removal Stemmers 21
3.1.4.1 Porter Stemming Algorithm 22
3.1.4.2 Lovins Stemmer 23
3.1.4.3 Paise Or Husk Stemming Algorithm 25

CHAPTER 4 IMPLEMENTATION OF N-GRAM STEMMING 27
4.1 Document Processing 27
4.2 Algorithm and Pseudo Code 29
4.2.1 Document processing code 29
4.2.2 Algorithm for n-gram stemming 32
4.3 Results 35

CHAPTER 5 CONCLUSION AND FUTURE WORK 39

BIBLIOGRAPHY 40

VITA 42

www.manaraa.com

LIST OF FIGURES

Figure 1 Retrieval Process 4
Figure 2 Inverted Indes Of Collection 6
Figure 3 Taxonomy Of Stemming Algoithms 17
Figure 4 List Of Tokens 27
Figure 5 Screen Shot of Test document 35
Figure 6 Screen Shot of Significant Terms 36
Figure 7 Screen Shot After Stop Word Removal 37
Figure 8 Screen shot of Maximally Connected Components 38

www.manaraa.com

ACKNOWLEDGEMENTS

I would like to take this opportunity to express the appreciation to my committee chair, Dr. Kazem Taghva for all his support and guidance through every stage of this thesis research. Without his guidance and persistent help, completion of this thesis would not have been possible.

I am very thankful to my graduate coordinator Dr. Ajoy K Datta for his help and invaluable support during my masters program. I extend my gratitude to Dr. Laxmi P. Gewali and Dr. Venkatesan Muthukumar for accepting to be a part of my committee. A special thanks to Ms.Lee for her help during my TA work. I would also like to take this opportunity to extend my gratitude to the staff of computer science department for all their help.

I would also like to extend my appreciation towards my parents and my friends for always being there for me through all phases of my work, for their encouragement and giving me their invaluable support without which I would never be where I am today.

www.manaraa.com

CHAPTER 1

INTRODUCTION

Information retrieval (IR) is a process of finding the material of an unstructured nature that satisfies information needed from within large collections of data. Stemming is one of the tools used in information retrieval to overcome the vocabulary mismatch problem. Stemming is a process of reducing words to their stem and is used in Information retrieval to reduce the size of index files and to improve the retrieval effectiveness. Idea here is to improve recall by automatically handling word endings by reducing the words to their word roots, at the time of indexing and searching. It is usually done by removing any suffixes and prefixes from index terms before the assignment of the term.

This thesis starts with understanding some of the basic information retrieval models and stemming algorithms followed by clustering of related pairs of words in the documents based on their character structure using an association measure. Association measure used here is dice coefficient. The collection which has been used here is NLP collection. This thesis have implemented one of the stemming algorithms called N-gram stemming and clustered the related pairs of words .The same experiment has been done by George W Adamson and Boreham in 1970 on a sample of words taken from chemical database.

www.manaraa.com

The output of the experiment was 90 percent of the related word pairs formed were correct. But when the experiment is carried on a very large data set the output was 60 percent of related word pairs.

1.1 Thesis Structure

This Thesis is organized into five chapters including the introduction chapter. Chapter 2 presents the Information retrieval chapter 3 gives details about stemming and types of stemming algorithms. Chapter 4 presents implementation details and experimental results of this thesis. Chapter 5 concludes this thesis by giving a brief description about future proceedings.

www.manaraa.com

CHAPTER 2

INFORMATION RETRIEVAL

Information retrieval (IR) is defined as 'finding material of an unstructured nature that satisfies information needed from within large collections' [1]. In other words, it is the science of searching for documents which contain the information required. The emergence of computers had made the task of storing large amounts of information easy. In 1950, the field of information retrieval (IR) was born, since finding the information that is useful and required from such collections had become essential [2].Information retrieval is fast becoming the dominant form of information access, overtaking traditional database style searching. In information retrieval, we will find those items that match the request partially and then filter them to find the best matched items [3]. A typical information retrieval system would look like in the figure below [5]. In an Information Retrieval Engine retrieval starts by the user entering the query to find documents that match required criteria. Before the retrieval process is started, a text model is developed from the document collection by performing text operations such as removing stop words and stemming. The text model is then used to build an index. Well Known models in information retrieval are Boolean model, Vector space model, co-ordinate matching, probabilistic model, language model.

www.manaraa.com

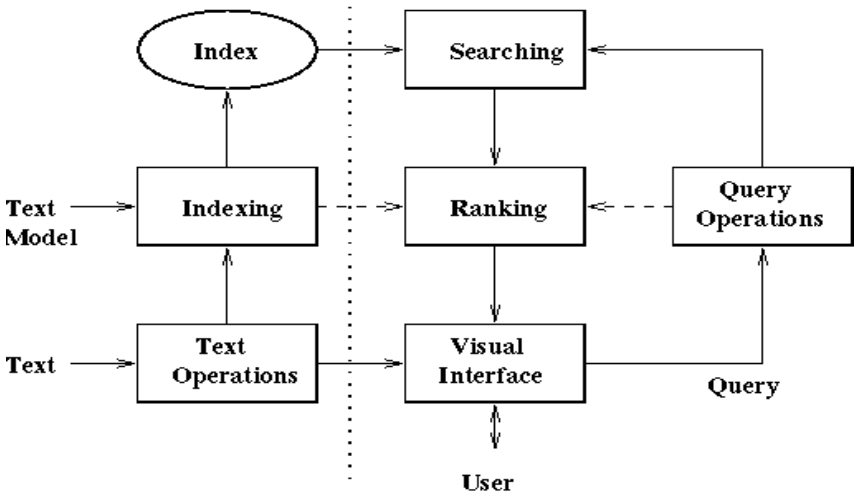

Figure 1. Retrieval Process

2.1 Boolean Retrieval

The *Boolean retrieval model* is a model for information retrieval in which any query is formed in the form of a Boolean expression of terms, that is terms are combined with the operators AND, OR, and NOT. In this model documents are represented by keywords or index terms. A document is considered to be relevant and retrieved if the index terms in the document satisfy the logical expression in the request. Users request is processed using inverted index file which is built for the collection. For each term in the query, the index is searched and the corresponding posting for the term is retrieved. Posting contains the list of documents in which the respective term occurs [1].

Once all the postings for the terms in the query are retrieved, they are merged based on the operator given in the query.

www.manaraa.com

For example 1:

S1= {A,B,C,D}

S2= {D,E,F,G}

S3= {G,I,J,K}

A,D,H: Index Terms

Q= A^D^~G

S1 is retrieved because S1 is true implies Q is true.

S2 and S3: Not Retrieved.

Example 2:[1]

Consider a small collection of four documents

Document ID	**Text**
Doc 1	breakthrough drug for schizophrenia
Doc 2	New schizophrenia drug
Doc 3	New approach for treatment of schizophrenia
Doc 4	New hopes for schizophrenia patients

Table 1. Document collection of four documents

The inverted index for the collection is build as shown in the figure below; in the inverted index document frequency of each term is stored. This information is used to minimize the amount of temporary memory space during query processing. In the figure, the left side shows all the terms which is also called as dictionary and the right hand side shows

www.manaraa.com

the postings. The core step in indexing is sorting the list of terms alphabetically.

Let us consider the following Boolean query and see the result.

Example User Boolean Query: Schizophrenia AND drug result.

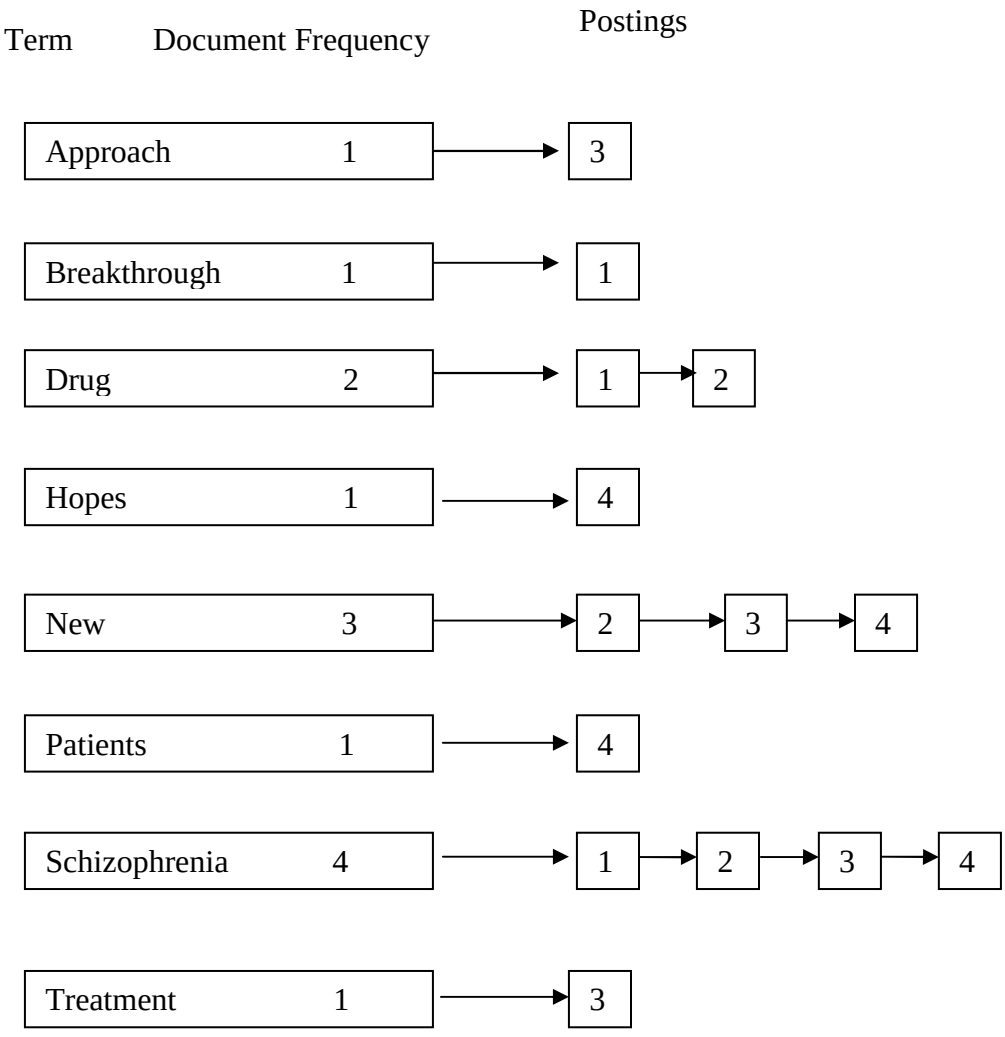

Figure 2. Inverted Index of collection

www.manaraa.com

In this case, the first term hence would be approach, then postings for the terms in the query will be loaded in to the memory. The postings of the remaining terms are compared against the posting in the memory. Since, it is a conjunctive query; the final result must be the list of documents which has all the terms in the query. In this case, the result is Doc 1 and Doc 2 because it contains both the words drug and schizophrenia.

Extended Boolean retrieval models can be built by adding additional operators other than AND, OR and NOT, such as proximity operators which gives how close two terms specified in the query can occur in the document. The main limitation of the Boolean retrieval model is its incapability to rank the result and to match documents that do not contain all the keywords of the query. In addition, more complex requests become very difficult to formulate. The vector space retrieval model addresses these issues.

2.2 Co-ordinate Matching

In this model, Requests are also a set of index terms. Documents that contain more number of terms in the query are given more importance than documents which contain few or none of them. Here we are calculating the inner product of query and each document both represented in form of n-dimensional vectors, where n is the number of terms in the index and then taking the result as the similarity measure.

www.manaraa.com

The similarity measure between the query and document in this type of retrieval model is represented as follows [4]

M (Q, Dd) = Q.Dd

For example, if we consider the same document collection given in Table 2.1.1 and a query "new drug". The vector representation of documents and sample query are given in the table below.

Doc ID	**Approach**	**Breakthrough**	**Drug**	**Hopes**	**New**	**Patients**	**schizophrenia**	**Treatment**
Doc 1	0	1	1	0	0	0	1	0
Doc 2	0	0	1	0	1	0	1	0
Doc 3	1	0	0	0	1	0	1	1
Doc 4	0	0	0	1	1	1	1	0
Query	0	0	1	0	1	0	0	0

Table 2. Vector representation of document collection and sample query

www.manaraa.com

Now, we can calculate the inner product of query and each document as follows:

M (new drug, Doc1) = (0, 0, 1, 0, 1, 0, 0, 0). (0, 1, 1, 0, 0, 0, 1, 0) = 1

M (new drug, Doc2) = (0, 0, 1, 0, 1, 0, 0, 0). (0, 0, 1, 0, 1, 0, 1, 0) = 2

M (new drug,Doc3)= (0, 0, 1, 0, 1, 0, 0, 0).(1, 0, 0, 0, 1, 0, 1,1) =1

M (new drug, Doc4) = (0, 0, 0, 0, 1, 0, 0, 1). (0, 0, 0, 1, 1, 1, 1, 0) = 1.

For this example query, the coordinate matching ranking is Doc2 > Doc1= Doc2 = Doc3 = 1.

The best feature of co-ordinate matching retrieval model is that it is very simple and straight forward as all the required information is in the inverted index. Also, in simplest way possible it introduces ranking, which means that it gives the result to the user's query in form of list of documents, the document with most of the query terms at the top. But, it has three notable drawbacks which are listed below [4]

1. Term frequency is not taken in to consideration, that is, in vector representation we just note if the term is "present" or "not present" using binary notation.

2. Term scarcity defines how important the term might be in describing the document, which is also not taken in to consideration.

www.manaraa.com

3. Long documents might always top the retrieval list since they are likely to have most of the query terms when compared to small documents.

To overcome first drawback, we can include the with-in document frequency ($f_{d,t}$) in the vector representation of documents. This will change the inner product similarity formulation as given below. [4]

$$M(Q, Dd) = Q.Dd = \sum_{t \in Q} w_{q,t} \cdot w_{d,t}$$

Where $w_{d,t}$ is the document-term weight for term t in document d. Similarly, $w_{q,t}$ is the weight for query vector.

For the second problem, the weight of the term ($w_{d,t}$) has to be reduced if it appears in many documents. This can be done by incorporating "Inverse document frequency" in to the term weight, which gives more importance or weight to the terms which occur less frequently in the documents and vice versa. Now, weight of the term, w_t can be calculated as $w_t = \frac{1}{f_t}$

Where f_t is the number of documents in which term t occurs. Now, $w_{d,t}$ can be calculated as [4]

$$w_{d,t} = f_{d,t} \times w_t$$

Assigning document-term weights is called TF×IDF rule. There are many variant methods available in the literature for calculating document-term weights with different interpretations for relative term

www.manaraa.com

frequency and inverse document frequency. One can choose which one to use based on a particular situation.

The last problem can be removed by taking the length of the document, which is count of the terms it contains in to consideration.

2.3 Vector Space Model

Representing a set of documents as vectors in a vector space is known as Vector space model. In this model each term t is considered as a dimension. A document d can be represented by the weight of each dictionary term.

$V(\vec{d}) = (W(t1,d), W(t2,d) \ldots W(tn,d))$

In this model query is also a vector representation of keywords in query and also has corresponding weights denoting the importance of respective keywords in the query. To assign a numeric score to a document for a query, this model measures the similarity between the query vector and document vector. Cosine angle is used as a similarity measure between the vectors (cosine angle has a property 1 for identical vectors and 0 for orthogonal vectors). As an alternative it can use the inner product between the vectors as a similarity measure.

If all the vectors are of unit length, then the cosine of the angle between two vectors is same as their dot-product. The cosine rule for ranking the documents is given below [4].

$$\text{Cosine}(Q, D_d) = \frac{1}{W_q W_d} \sum_{t=1}^{n} w_{q,t} . w_{d,t}$$

www.manaraa.com

Where, $W_q = \sqrt{\sum_{t=1}^{n} w^2_{q,t}}$ and $W_d = \sqrt{\sum_{t=1}^{n} w^2_{d,t}}$

In the above equations, $w_{q,t}$ and $w_{d,t}$ denote the weights of the terms in the query and the document respectively. There are many different algorithms to weigh these terms and which one to choose depends on the characteristics of the collection. Once the cosine measures between the vectors are calculated results are displayed to the user in descending order of document's cosine measure values.

One of the man disadvantage of the vector space model is it assumes the independence of index terms.

2.4 Probabilistic Retrieval Model

Probabilistic models are based on the general principle that documents in the collection should be ranked by decreasing probability of their relevance to a query. This is called as Probabilistic ranking principle. Since true probabilities are not available to information retrieval system probabilistic information retrieval models estimate the probability of relevance of documents for a query. It is an alternative model for query optimization.

Two main parameters in this model are P(REL) and P(NREL) i,e probability of relevance and probability of non-relevance of a document . Probability that a document d is relevant is given by

P(REL/d) = (P(REL) * P(d/REL))/P(d)

To avoid the expansion of P(d) we take the log odds

www.manaraa.com

$$\log \frac{P(REL/D)}{P(NREL/D)} = \log \frac{P(D/REL)P(REL)}{P(D/NREL)P(NREL)}$$

In the above equation P(REL) and P(NREL) are just the scaling factors you can remove them from the above formulation.

$$\log \frac{P(D/REL)}{P(D/NREL)}$$

In probabilistic retrieval model we classify the document d as relevant if

P(D/REL)P(REL)>P(D/NREL)P(NREL)

So P(D/REL) can be written as a product of each term's probabilities: [2]

$$P\ (D/REL) = \prod_{ti \in Q,D} P(ti/REL).\prod_{tj \in Q,D}(1 - P\ (tj/REL))$$

The above equation uses two probabilities; one is the probability of presence of term t_i in relevant documents set. The other is the probability of absence of term t_j in relevant documents set. Here, we consider all the terms which are common to the query and the document.

Substituting the value of P(D/REL) in the log of odds equation and also removing constant values for a given query, we get the following ranking function. For further simplification we denote $P(t_i/REL)$ as p_i and $P(t_i/NREL)$ as q_i [2].

$$\log \prod_{ti \in Q,D} \frac{pi.(1-qi)}{qi.(1-pi)}$$

2.5 Language Model

In Language modeling approach to Information Retrieval a document is a good match to a query if the document model generates the query, which

www.manaraa.com

happens if the document contains the query words [1].In Probabilistic approach we model the probability of relevance of a document d to a query q but in language modeling approach, from each document d a probabilistic language model M_d is build and the documents are ranked based on the probability of the model generating the query $P(q/M_d)$.

Document model generating a query is model of a language that can be used either to recognize or generate strings. A language model is a function which gives the probability measure over strings drawn from vocabulary. A model which estimates each term independently without considering any condition is called as unigram language model.

$P_{uni}(t1t2t3t4)=P(t1)p(t2)p(t3)p(t4)$

Languages models which conditions on the previous terms are called bigram language models.

$P_{bi}(t1t2t3t4)=p(t1)p(t2/t1)p(t3/t2)p(t4/t3)$

There are some more complex grammar based language models used for speech recognization, spelling correction and so on.

The Query Likelihood model:

It is the basic methods for using language models .In this model documents are ranked by P(d/q).

By bayes rule,

$P(d/q)=(p(q/d)p(d))/p(q)$

In the above equation p(d) and p(q) are eliminated so the results are ranked by p(q/d) i.e. the probability of query q under a language model

www.manaraa.com

derived from d. The maximum likelihood estimate (MLE) of term t, given the model is given by [1].

$$P\hat{}_{ml}(t/\ M_d) = \frac{termfrequency\ in\ document(tft,d)}{total\ number\ of\ tokens(dld)}$$

The ranking formula for each document which is $P(Q/\ M_d)$ can be calculated using the following [1]:

$$P\hat{}\ (Q/\ M_d) = \prod_{t \in Q} P\hat{}ml(t/\ Md)$$

The symbol (^) suggests that the model is estimated. If the term did not occur smoothing weights are assigned to $P\hat{}_{ml}(t/\ M_d)$. Usually a minimal value is assigned that means that it might still be possible for the term to occur. In other words,

if $tf_{(t,d)} = 0$, then we assign

$$P\hat{}_{ml}(t/\ M_d) = \frac{cft}{cs}$$

Where cf_t is term count in the collection and cs is the total number of tokens in the collection. There are a variety of smoothing techniques available for overcoming this practical problem of assigning zero weights [1].

www.manaraa.com

CHAPTER 3

STEMMING ALGORITHMS

Information retrieval is process of retrieving the documents to satisfy the users need for information. The user's information need is represented by a query, the retrieval decision is made by comparing the terms of the query with the terms in document itself or by estimating the degree of relevance that the document has to the query. Words in a document may have many morphological variants .These morphological variants of words have similar semantic interpretations and can be considered as equivalent for the purpose of IR applications. For this reason, a number of so-called stemming Algorithms, which reduces the word to its stem or root form have been developed. Thus, the key terms of a query or document are represented by stems rather than by the original words. Stemming reduces the size of the index files and also improves the retrieval effectiveness. Fig 3.1 shows the taxonomy of stemming algorithms.

www.manaraa.com

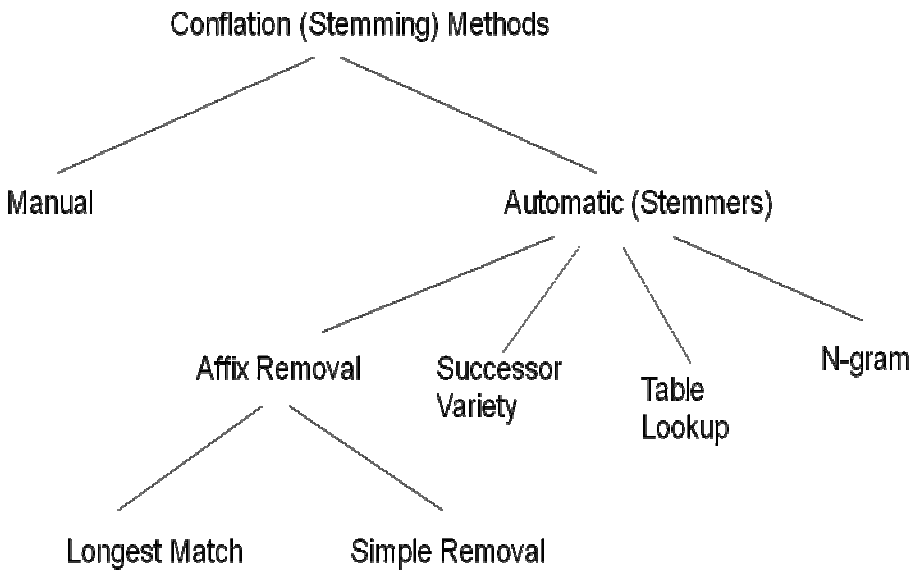

Figure 3. Taxonomy of stemming algorithms

There are four automatic approaches. Affix removal algorithms removes affixes or prefixes from terms leaving a stem. Successor variety stemmers use the frequencies of letter sequences in the text as the basis for stemming .N-gram method conflates the terms based on the number of digrams or n-grams they share .Correctness, retrieval effectiveness and compression performance judges the stemmers. There are two was a stemming can be incorrect over stemming and under stemming. When a term is over stemmed too much of the stem is removed. Over stemming may cause unrelated terms to be conflated. Under stemming is removal of too little of a term and will make the related terms from being conflated.

www.manaraa.com

3.1 TYPES OF STEMMING ALGORITHMS

3.1.1 Table Look Up Approach

One method to do stemming is to store a table of all index terms and their stems .Terms from the queries and indexes could then be stemmed via lookup table, using b-trees or hash tables. Such lookups are very fast, but there are problems with this approach. Firstly there is no such data for English even if there were they may not be represented because they are domain specific and require some other stemming methods. Second issue is storage overhead.

3.1.2 Successor Variety

Successor variety stemmers are based on the structural linguistics which determines the word and morpheme boundaries based on distribution of phonemes. Successor variety of a string is the number of characters that follow it in words in some body of text. For example consider a body of text consisting of following words.

Able, ape, beatable, finable, read, readable, reading, reads, red, rope, ripe.

Let's determine the successor variety for the word read. First letter in read is R. R is followed in the text body by 3 characters E, I, O thus the successor variety of R is 3. The next successor variety for read is 2 since A, D follows RE in the text body and so on .Following table shows the complete successor variety for the word read.

www.manaraa.com

Prefix	Successor Variety	Letters
R	3	E,I,O
RE	2	A,D
REA	1	D
READ	3	A,I,S

Table 3. Successor variety for word read

Once the successor variety for a given word is determined then this information is used to segment the word. Hafer and Weiss discussed for ways of doing this.

1. Cut Off Method: Some cutoff value is selected and a boundary is identified whenever the cut off value is reached.

2. Peak and Plateau method: In this method a segment break is made after a character whose successor variety exceeds that of the characters immediately preceding and following it.

3. Complete word method: Break is made after a segment if a segment is a complete word in the corpus.

4. Entropy Method: In this method $|D_{\alpha i}|$ is the number of words in a text body beginning with the i length sequence of letters α. $|D_{\alpha ij}|$ is the

www.manaraa.com

number of words in $D_{\alpha i}$ with the successor j. The probability that a member of $D_{\alpha i}$ has the successor j is given by .

The entropy of $D_{\alpha i}$ is given by $\frac{|D\alpha ij|}{|D\alpha i|}$

$$H_{\alpha i} = \sum_{j=1}^{26} -\frac{|D_{\alpha ij}|}{|D_{\alpha i}|} \cdot \log_2 \frac{|D_{\alpha ij}|}{|D_{\alpha i}|}$$

Using this method entropy for a word and its predecessors is determined then cut off value is selected and boundary is identified when cutoff value is reached.

3.1.3 N-Gram stemmers

This method has been designed by Adamson and Boreham. It is called as shared digram method. Digram is a pair of consecutive letters. This method is called n-gram method since trigram or n-grams could be used. In this method association measures are calculated between the pairs of terms based on shared unique digrams.

For example: consider two words Stemming and Stemmer

Stemming → st te em mm mi in ng

Stemmer →st tee m mm me er

In this example the word stemming has 7 unique digrams ,stemmer has 6 unique digrams, these two words share 5 unique digrams st, te, em, mm ,me. Once the number of unique digrams is found then a similarity

www.manaraa.com

measure based on the unique digrams is calculated using dice coefficient. dice coefficient is defined as

S=2C/(A+B)

Where C is the common unique digrams, A is the number of unique digrams in first word; B is the number of unique digrams in second word. Similarity measures are determined for all pairs of terms in the database, forming a similarity matrix, Once such a similarity matrix is available, terms are clustered using a single link clustering method.

3.1.4 Affix Removal Stemmers

Affix removal stemmers removes the suffixes or prefixes form the terms leaving the stem. One of the example of the affix removal stemmer is one which removes the plurals form the terms. Some set of rules for such a stemmer are as follows (Harman)

a) If a word ends in “ies” but not ”eies” or ”aies ”

Then “ies” -> “y”

b) If a word ends in “es” but not ”aes” , or ”ees ” or “oes”

Then “es” -> “e”

c) If a word ends in “s” but not ”us” or ”ss ”

Then “s” -> “NULL

www.manaraa.com

Stemmers which are currently in use are iterative longest match stemmers these are kind of affix removal stemmers developed by Lovins. In addition to Lovins iterative longest match stemmers have also been given by Salton, Dawson, Porter and Paice.

3.1.4.1 Porter Stemming Algorithm

The Porter stemmer was developed by Martin Porter in 1980. Porter stemming algorithm is a context sensitive suffix removal algorithm and is the most widely used of all the stemmers. The stemmer is divided into a number of linear steps that are used to produce the final stem. A consonant is a letter other than A, E, I, O, U and Y preceded by a consonant. A vowel is any letter that is not a consonant. A list of consonants greater than or equal to length one will be denoted by a C and a similar list of vowels by a V.

Any word can be represented by the single form; [C] (VC)m [V] Where the superscript m denotes m repetitions of VC and the square brackets [] denote the optional presence of their contents [6] The value m is called the measure of a word and can take any value greater than or equal to zero, and is used to decide whether a given suffix should be removed. All such rules are of the form S1 -> S2 means that the suffix S1 is replaced by S2 if the remaining letters of S1 will satisfy the condition.

The first step in the algorithm is the most complex and is separated into three parts in the original definition, 1a, 1b and 1c. The first part

www.manaraa.com

deals with plurals, for example sses -> ss and removal of s. The second part removes ed and ing, or performs eed where appropriate. The second part continues only if ed or ing is removed and transforms the remaining stem to ensure that certain suffices are recognized later. The third part simply transforms a terminal y to an i. The remaining steps in this stemmer contain rules to deal with different order classes of suffices, initially transforming double suffices to a single suffix and then removing suffices provided the relevant conditions are met.

3.1.4.2 Lovins Stemmer

The Lovins stemming algorithm is developed by Julie Beth Lovins in 1968. It is a context sensitive and single pass stemmer, which removes endings based on the longest-match principle. This stemmer utilizes many rules that are designed to overcome the most common exceptions. All endings are associated with the default exception that is every stem must be at least two letters long, which is designed to prevent the production of ambiguous stems. Other rules maintain one of the following conditions on the ending's removal,

i) Minimum length of a stem is increased by following ending's removal.

ii) Prevent removing of endings when certain letters are present in the remaining stem.

iii) Combination of the above restrictions.

www.manaraa.com

When developing the stemmer Lovins described that the rule that can be generalized to apply in numerous situations is the most desirable form of context sensitive rule. It was discovered that Few examples of such rules could be found during the development of the stemmer. Number of special cases exist for each ending that cause erroneous stems to be produced, these are often unique to the ending and number of rules would have to be developed that would prevent errors. This process would require large amounts of time and data, and would lessen the improvements in performance over time. For this reason it was decided to deal with the more obvious exceptions and to hopefully limit the number of errors that remain unaccounted for in the exception list.

This algorithm has two phases. The stemming phase and recording phase. Stemming phase is been discussed above and includes the removal of endings and the testing of associated exceptions among other steps. The second part of the algorithm is the recoding phase. The term spelling exception is used to cover all the situations in which a stem may be spelled in more than one way. The majority of these exceptions that occur in English are due to “Latinate derivations” such as matrices and matrix. Other types of exceptions occur that can be attributed to differences in British and American spellings, such as analysed and analyzed, or to basic inflexion rules that cause the doubling of certain consonants when a suffix is added. Lovins proposed two ways to deal with this problem which are called partial and recording matching.

www.manaraa.com

3.1.4.3 Paise Or Husk Stemming Algorithm

It was developed by Chris Paice Has developed the Paice/Husk stemmer with the assistance of Gareth Husk in 1990. This stemmer is a conflation based iterative stemmer.

The stemmer utilizes a single table of rules, each of which may specify the removal or replacement of an ending. This technique of replacement is used to avoid the problem of spelling exceptions by replacing endings. This stemmer does this without a separate stage in the stemming process, i.e. no recoding or partial matching. This helps to maintain the efficiency of the algorithm. The rules are indexed by the last letter of the ending to allow efficient searching and are of the following form

i) An ending of one or more characters, held in reverse order

ii) An optional intact flag '*'

iii) A digit specifying the removal total (zero or more)

iv) An optional append string of one or more characters

v) A continuation symbol, '>' or '.'

This algorithm has four main steps detailed below

1.Select relevant section: Inspect the final letter of the term and, if present, consider the first rule of the relevant section of the rule table.

2. Check applicability of rule: If final letters of term do not match rule, or

www.manaraa.com

intact settings are violated or acceptability conditions are not satisfied go to step 4.

3. Apply Rule: Remove or reform ending as required and then check termination symbol, and either terminate or return to step 1.

4. Look for another rule: Move to the next rule in table, if the section letter has changed then terminate, else go to step 2.

Stemmers are used to conflate terms to improve retrieval effectiveness and /or to reduce the size of indexing file. Stemming will increase recall at the cost of decreased precision. Stemming can have marked effect on the size of indexing files, sometimes decreasing the size of file as much as 50 percent.

www.manaraa.com

CHAPTER 4

IMPLEMENTATION OF N-GRAM STEMMING

4.1 Document Processing

Initially all the Training Documents are tokenized. Tokenization is the process of breaking parsed text into pieces, called tokens [21]. During this phase text is lowercased and punctuations are removed. For example consider the sentence "Although there was inflation, at least the economy worked," from a document that belong to category Trade it is tokenized as shown in Table 4.3.

although
There
Was
Inflation
At
Least
The
Economy
Worked

Figure 4. List of tokens.

www.manaraa.com

Next step after tokenization is removing stop words. Common words such as 'are', 'the', 'with', 'from' etc. that occur in almost all documents, does not help in deciding whether a document belongs to a category or not. Such words are referred as stop words. So, these words can be removed by forming a list of stop words. This thesis works on a total of 416 stop words.

Before removing stop words there are a total of 52034 terms in the training documents, but after removing stop words there are reduced to 27910 terms including duplicates. Thus, 24124 words are removed which appeared to be of little value saving both space and time. Once stop words are removed, next step performed is stemming.

In this thesis I am using n-gram stemmers. It is a shared digram method. We call it as n-gram stemmers because we can also use trigrams or n-grams instead of digrams. In this method association measures are calculated between pairs of words in the document based on shared unique digrams .once the unique digrams for a pair of words have been identified a similarity measure based on them is computed. The similarity measure used was dice coefficient which is defined as

$$S=2C/ (A+B)$$

Where C is the common unique digrams in the word pair

A is the unique number of digrams in first word

B is the unique number of digrams in second word.

www.manaraa.com

Such similarity measures are determined for all pairs of words in the documents forming a similarity matrix. Once the similarity matrix is determined words are all clustered using maximally connected components.

4.2 Algorithm and Pseudo Code

4.2.1 Document processing code

Document processing is done as described in section 4.1. following is the pseudo code for document processing .

```
//File Tokenizing.java
class FileTokenizer
{
public static void main(String args[]){
        try{
        // Create the tokenizer to read from a file
        FileReader rd = new FileReader("test.txt");
        StreamTokenizer st = new StreamTokenizer(rd);

        // Prepare the tokenizer for Java-style tokenizing rules
        st.parseNumbers();
        st.wordChars('_', '_');
        st.eolIsSignificant(true);

        // These calls caused comments to be discarded
       st.ordinaryChar('.');
        st.ordinaryChar("");
        st.ordinaryChar('/');
        st.ordinaryChar('\');
        st.ordinaryChar('<');
        st.ordinaryChar('>');
        st.ordinaryChar(':');
        st.ordinaryChar('(');
        st.ordinaryChar(')');
       // st.ordinaryChar("");

          st.slashSlashComments(true);
         st.slashStarComments(true);
          st.lowerCaseMode(true);
          System.setOut(newPrintStream(new
FileOutputStream("tokenize.txt")));
```


www.manaraa.com

```
        // Parse the file
        int token = st.nextToken();
            while (token != StreamTokenizer.TT_EOF) {
            switch (token) {
                case StreamTokenizer.TT_NUMBER:
                // A number was found; the value is in nval
               int num =(int)st.nval;
         //javacSystem.out.println("  "+ num+" " );
}
//removestopwords.java
import java.lang.Object;
import java.io.*;
import java.lang.String;
import java.util.*;

class Removewords
{
public static void main(String args[])
{
try
{
  String[] STOP_WORDS =
    {

        "s", "t", "u", "v", "w", "x", "y", "z","$"
    };


int len=STOP_WORDS.length;
System.out.println(len);

//FileReader rd=new FileReader("list.txt");
FileReader fr=new FileReader("tokenize.txt");
StreamTokenizer st=new StreamTokenizer(fr);
//ArrayList<String> ar=new ArrayList<String>();
//ArrayList<String> ar1=new ArrayList<String>();


BufferedWriterout=new                    BufferedWriter(new
FileWriter("removestopwords.txt"));
 //System.setOut(new                       PrintStream(new
FileOutputStream("removestopwords.txt")));
String s;

int token=st.nextToken();
```


www.manaraa.com

```
int i=0,count,check;
String str,str1;
String output=null;
 while(token!=StreamTokenizer.TT_EOF)
  {


  str=st.sval;

   check=0;
for(i=0;i<len;i++)
{
  String t=STOP_WORDS[i];
if(t.equalsIgnoreCase(str))
 {
  check=1;
output=str.replaceAll(t,"WAY");
   }

}


if(check==1)
{

}
else
{
out.write(str);
 out.newLine();
}
token=st.nextToken();

}
out.close();

 fr.close();
  }
```


www.manaraa.com

4.2.2 Algorithm for n-gram stemming

a)Similarity Coefficient

The measure of association used here is dice coefficient[9].Let a and b be the number of digrams in word A and word B. c is the number of digrams common to A and B. Similarity coefficient is

$S_{AB}=2c/(a+b)$

This coefficient was choosen for the ease of computation.

b) Coefficients calculation

The coefficient of similarity between two words is computed as follows. First a digram string is generated for each word and stored. Comparison of string determines the number of co-occurring digrams. Multiple occurrence of the same digram is treated as distinct. Final count of total number of digrams in each word is required for the computation.

Once the similarity coefficients for all the words are obtained a similarity matrix is found, this is used to cluster the words by finding maximally connected words or strongly connected words in the documents.

The data set used in this thesis is NLP collection containing 11429 documents. After tokenization 479163 words are obtained. Form the words that are obtained we removed the stop words and obtained 13529 number of words.

www.manaraa.com

Pair wise similarity coefficients for each of these words have been found and a similarity matrix is obtained. All the words which had similarity coefficient as 0.6 or more are considered into the similarity matrix.

All the words which are semantically related are grouped together using the method of maximally connected components .In this experiment total clusters formed are 1999 out of which 1242 are appropriate, this 1242 clusters contain 5597 words in it. 130 words formed among them are inappropriate.

//pseudo code for n-gram stemming

```
LinkedList<Integer> current = new LinkedList<Integer>();
  if(k<words.length-1)
  al.add(k, current);
  int flag=0;
  for (int l = k + 1; l < words.length; l++)
  {
      if(b[l]==1)
          continue;
      String[] grams1 = Generatebigrams(words[k], 2);
      String[] grams2 = Generatebigrams(words[l], 2);
      int count = 0;

      for (int i = 0; i < grams1.length-1; i++)
      {
          for (int j = 0; j < grams2.length; j++)
          {
              if (!grams1[i].equals(grams2[j]))
                  continue;
              count++;
              break;
          }
      }
      float sim = (2.0F * (float) count)/ (float) (grams1.length +
grams2.length);
          if (sim > treshhold)
      {   current.add(l);
```


www.manaraa.com

```
                    g.addVertex(words[k]);
                    g.addVertex(words[l]);
                    g.addEdge(words[k],words[l]);
                    g.addEdge(words[l],words[k]);
                    b[l]=1;
                    flag=1;
                }
                }
    //pseudo code for finding maximally connected maximally connected
    components
String[] words = new String[wordList.size()];

        wordList.toArray(words);

        // loop and display each word from the words array

        for (int i = 0; i < words.length; i++)
        {
            System.out.println(words[i] + " " +i);

        }

        DirectedGraph<String,DefaultEdge> g=new
DefaultDirectedGraph<String,DefaultEdge>(DefaultEdge.class);

StrongConnectivityInspector sci=new StrongConnectivityInspector(g);
        java.util.List<java.util.Set<String>> re=sci.stronglyConnectedSets();
        System.out.println(re.size());
        System.out.println(re);

private static String[] Generatebigrams(String text, int gramLength)
    {
        // ArrayList grams=new ArrayList();
        List<String> grams = new ArrayList<String>();

        // System.out.println("Printing:"+text);
        int length = text.length();

        for (int i = 0; i < length - 1; i++)
        {
            String gram = text.substring(i, i + 2);
            // System.out.println(gram);
            grams.add(gram);
            // System.out.println(grams.get(i));
        }
        String[] g = new String[grams.size()];
```


www.manaraa.com

```
grams.toArray(g);
return (g);

}
```

4.3 Results

4.3.1 Screen Shots

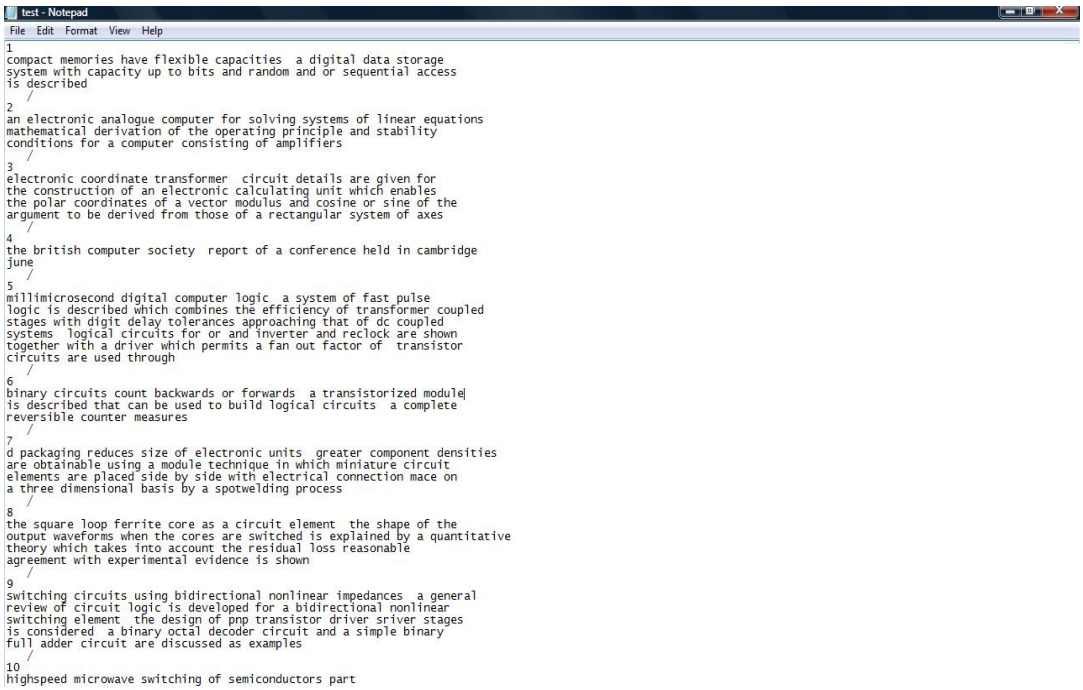

Figure 5. Screen shot of test document

www.manaraa.com

compact
memories
have
flexible
capacities
a
digital
data
storage
system
with
capacity
up
to
bits
and
random
and
or
sequential
access
is
described
an
electronic
analogue
computer
for
solving
systems
of
linear
equations
mathematical
derivation
of
the
operating
principle
and
stability
conditions

Figure 6. Screen shot of significant terms.

www.manaraa.com

compact
memories
flexible
capacities
digital
data
storage
system
capacity
bits
random
sequential
access
electronic
analogue
computer
solving
systems
linear
equations
mathematical
derivation
operating
principle
stability
conditions
computer
consisting
amplifiers
electronic
coordinate
transformer
circuit
details
construction
electronic
calculating
unit
enables
polar
coordinates
vector
modulus

Figure 7. Screen shot after stop word removal.

www.manaraa.com

[shock]
[shoes, shoed]
[vortex]
[marcoussis]
[shafts, shaft]
[true]
[ahmedabad]
[fused]
[whih, whic]
[aligned]
[venus]
[microbarograph, microbaroms]
[pool, poor]
[baileys, bailey]
[cases]
[lunar]
[thresholds, threshold]
[english]
[valued, values, valuex]
[seed, sees, seen]
[kingdom, king]
[shoul, shoud]
[leg, clegg]
[spike, spikes]
[seat, seac]
[aeronautics, aeronautical]
[aircraft, aircrew]
[depths, depth]
[silent, tensile]
[howarth, howard]
[sunspots, sunspot, sunsport]
[clear, clean]
[substituting, substituted]
[volume, volumes]
[favourable, favourably]

Figure 8. Screen shot of maximally connected words.

www.manaraa.com

CHAPTER 5

CONCLUSION AND FUTURE WORK

The main objective of this thesis is to examine the working of stemming algorithms and implement one of the stemming algorithms. N-Gram stemming was implemented among the different stemming algorithms discussed in chapter 3. Based on the results and evaluation performed on n-gram stemming algorithm, we can conclude that this stemming works really well for a small set of documents. This method successfully grouped individual words into semantically related clusters and appears to be a plausible technique for improving IR performance.

This thesis concentrates on N-gram stemming algorithm. Other algorithms can be implemented and the performance between them can be compared over a larger collection of data. Further evaluation of the n-gram stemming can be done by using different similarity coefficients and by using different clustering methods to cluster the semantically related set of words in the document.

www.manaraa.com

BIBLIOGRAPHY

1. Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze, 'Introduction to Information Retrieval', Chapters 1, 6, 8, 9, 11 & 12, Cambridge University Press, 2008.http://nlp.stanford.edu/IR-book/html/htmledition/irbook.html.

2. Amit Singhal, 'Modern Information Retrieval: A Brief Overview', IEEE Data Engineering Bulletin, Volume 24, pages 35-43, 2001.

3. C. J. Van Rijsbergen, 'Information Retrieval', Second Edition, Chapters 1, 6 & 7, Information Retrieval Group, University of Glasgow, London: Butterworths, 1979.

4. Ian H. Witten, Alistair Moffat, and Timothy C. Bell, 'Managing Gigabytes', Second Edition, Chapter 4, Morgan Kaufmann Publishers, Inc, San Francisco, May 1999.

5. Baeza Yates, Berthier Riberio Neto, 'Modern Information Retrieval', Chapter 1, Addison Wesley, Addison Wesley Longman, 1999. http://people.ischool.berkeley.edu/~hearst/irbook/1/node2.html.

6. Frakes, W. B ' Information Retrival',Second Edition, Chapters 8, Stemming Algorithms.

7. Martin Porter, 'The Porter Stemming Algorithm', Jan 2006. http://tartarus.org/~martin/PorterStemmer/.

8. www.comp.lancs.ac.uk.

9. Information retrieval group by Keith van Rijsbergen, Test Collections. http://ir.dcs.gla.ac.uk/

10. Kiran Pai, 'A simple way to read an XML file in Java', 2002. http://www.developerfusion.com/code/2064/a-simple-way-to- read-an-xml-file-in-java/

11.HappyCoders,'TokenizingJavasourcecode'(n.d.) http://www.java.happycodings.com/Core_Java/code84.html

12. Adamson,G, and Boreham 1974 "The Use Of An association Measure Based on Character Structure to Identify Semanticaaly Related Pairs Of Words and Document Titles," Information storage and Retrieval,10,253-60.

www.manaraa.com

13. Frakes,W.B 1982.Term Conflation For Information Retrieval,Doctoral Dissertation,Syracuse University.

14. Hull, D. Stemming Algorithms: A Case Study for Detailed Evaluation. In *Journal of the American Society for Information Science*.

15. VanRijsbergen, C.J. *Information Retrieval*, Butterworths. London, 1979.http://dcs.glasgow.ac.uk/keith

16. Frakes, W.B. Stemming Algorithms. In Frakes, W.B. & Baeza-Yates, R.(Eds.), *Information Retrieval, Data Structures and Algorithms*. New Jersey:Prentice-Hall, pp. 131-160, 1992

17. R.R Sokal and P.H.A.Sneath: Principals of Numerical Taxonomy.Freeman ,SanFrancisco.

www.manaraa.com

VITA

Graduate College
University of Nevada, Las Vegas

Savitha Kodimala

Degrees:
Bachelor of Technology in Computer Science, 2008
Jawaharlal Nehru Technological University

Master of Science in computer science, 2010
University of Nevada Las Vegas

Thesis Title: Study of Stemming Algorithms

Thesis Examination Committee:
Chair Person, Dr. Kazem Taghva, Ph.D.
Committee Member, Dr. Ajoy K. Datta, Ph.D.
Committee Member, Dr. Laxmi P. Gewali, Ph.D
Graduate College Representative, Dr. Muthukumar Venkatesan, Ph.D.

www.manaraa.com