International Journal of Modern Engineering Research (IJMER)

www.ijmer.com Vol.2, Issue.3, May-June 2012 pp-945-948 ISSN: 2249-6645

Index based Information Retrieval System

Ambesh Negi[1], Mayur Bhirud[1], Dr. Suresh Jain[2], Mr.Amit Mittal[3]

[1]*PG Scholar, IET DAVV,.Indore*

[2]*Director, KCBTA, Indore*

[3]*Assistant professor, IET, Indore*

Abstract: **Information retrieval system based on keyword searching deals with a very large search space as documents to be searched can be of any length and thus time to search in a whole document is also proportional to length of documents i.e. number of words in all documents.**

By reducing this large search space search time can also be reduced. In this paper we are proposing a method which reduces the search space with the help of indexing that uses concept of stemming and knowledge of stopwords. Indices are created for single terms and phrases both so that a single concept whether it is represented by a word or more than one word can be treated as required. Our search method uses ontology to incorporate domain knowledge while searching and thus improves the recall.

Keywords: *Information Retrieval, Indexing, Keyword searching, query , search space, ontology.*

I. INTRODUCTION

Information retrieval deals with the storage and representation of knowledge and the retrieval of information relevant to a specific user problem. Information retrieval systems respond to queries which are typically composed of a few words taken from a natural language. The query is compared to document representations which were extracted during the indexing phase. The most similar documents are presented to the users who can evaluate the relevance with respect to their information needs and problems.

Many previous retrieval systems based on keyword searching represent documents and queries by the words they contain and base the comparison on no. of words they have in common. The more the words the query and document have in common, the higher the document is relevant. This refers to as coordination match but there are few problems in this approach. first is that a word in a document can appear in many lexical forms for an example word information can have multiple forms as inform, informed ,informing etc. in the keyword matching approach if you want to search word inform , then it should be spelled same although informed and informing could be of use. Second problem is that query words have to be matched with a bag of words* representing their respective documents which is very cumbersome task. Another problem is that if words in the query do not appear in the documents there will be a no match situation arise so we need to somehow increase our recall**.

These problems can be solved by removing unuseful words from search space which are called stopwords. Using a proper stemming algorithm can solve multiple form problems. Also using some domain knowledge and ontology we could add recall to our system by query expansion.

With this idea in mind, we designed a system which uses a standard stemming algorithm, some ontology using domain knowledge and a proper retrieval approach that performs a ranked retrieval on documents based on user query. Also the retrieval is done term based and phrase based separately as a phrase can also be an important term consisting of multiple words.

* Bag of words contains almost each word of document except only stop words

** Recall means fraction of relevant documents that are retrieved.

The structure of this paper is as follows. A brief review of previous research is presented in Section 2, followed by Proposed method in section 3. Section 4 by a description of results. Finally, Section 5 covers conclusions and future work.

II. LITERATURE SURVEY

Many keyword based search has been performed where pros and cons of the approach and integration of it with many of the well known approaches has been carried out some of the important are discussed here.

Rajasekar Krishnamurthy Sriram Raghavan Shivakumar Vaithyanathan Huaiyu Zhu [1] address the problem of building a retrieval system that is specifically targeted to answer search tasks that fit the above description (hereafter referred to as precision-oriented search tasks).

Baid, A.; Rae, I.; AnHai Doan; Naughton, J.F.[2] gave basic idea which is to produce answers as in today's KWS systems up to the time limit, then show users these answers as well as query forms that characterize the unexplored portion of the answer space. Finally, they present some

www.ijmer.com Vol.2, Issue.3, May-June 2012 pp-945-948 ISSN: 2249-6645

preliminary experiments over real-world data to demonstrate the feasibility of the proposed solution approach.

In the paper Research on Ontology-Driven Information Retrieval Stein L. Tomassen Department of Computer and Information Science, Norwegian University of Technology and Science, NO-7491 Trondheim, Norway examines how ontologies can be efficiently applied to large-scale search systems for the web. We describe how these systems can be enriched with adapted ontologies to provide both an in-depth understanding of the user's needs as well as an easy integration with standard vector-space retrieval systems. The ontology concepts are adapted to the domain terminology by computing a feature vector for each concept.

Many researchers have compared the effectiveness between manual and automatic indexing techniques. Manual indices were often presumed to be better than machine generated indices. However, it has been demonstrated that both indexing techniques are equally effective for text retrieval [Salton][3]. The retrieval performance were also showed positive improvement if both techniques were combined compared to individual indexing [Rajashekar & Croft].[4]

III. PROPOSED METHOD

We proposed here the term based and phrase based text retrieval process consisting of following phases:

Indexing:

Preprocessing of document: Raw documents must be converted into bag of terms representation. These expressions are some times called document representatives. To make these representatives for each document we first collect the words and create the file containing words except the stop words, then we stem the words with in a file thus we have all the terms important to our search in their root forms.

After this we count the frequency of each word and the word having frequency above a threshold (based on a formula consisting file size) is selected as an index term. Collection of all such terms creates our index table (document representative) for that document.

Same process will be repeated for phrase based search but the difference is that all work will be done on phrases identified by a phrase identification program which identify phrases by counting their frequencies; phrase having a suitable count will be selected for processing. Thus an index table will be generated containing phrases as its indices.

Query formulation:

First step here is to expand the query based on domain knowledge (in our case computer science subjects) stored in the form of ontological structure as a tree. Query words are searched in the tree and their parent; children and siblings words are added in the query. Now what is the significance of adding these words? Answer is that they represent similar (in case of siblings) and relative (in case of parent, child) concepts which would increase our recall while searched against documents.

Second step is to apply the Preprocessing approach described above to the query, thus we reduce our query in the form identical to our document representative.

Same process will be repeated for phrase based search as we use a separate phrasal query to search against document representatives based on phrases.

Comparison:

The system compares the user query to the stored document representatives, and makes a classification decision about which documents to retrieve and in what order. Documents or parts of documents are displayed. Before searching user can select whether he wants to expand the query using tree or not.

This comparison is carried out on the basis of matrix multiplication approach in which document representatives are converted into an id by term matrix(where no. of terms equal to all terms together in all documents and id means file id assigned initially) and a matrix is generated for query terms. Multiplication of both provides necessary result to identify which document is more relevant to the query. Mathematically it can be shown as:

Consider there are 2 documents (i and j) represented as:

Doc (i) = (Term (i1),Term(i2)……Term(ik)
Doc (j) = (Term (j1),Term(j2)……Term(jl)

Where k and l are no. of terms in respective documents.

So, all terms for all documents together can be represented as =

[Term (i1), Term (i2)……Term (ik) U (Term (j1), Term (j2)……Term (jl)]

- [Term (i1), Term (i2)……Term (ik) ∩ (Term (j1), Term (j2)……Term (jl)]

= [(Term (1), Term (2)……Term (n)]

i.e. Term(1)….Term(n)= all distinct terms of both documents i and j.

Our comparison is based on weighted values and implication of inverted document frequency (IDF):

- Weight is how many times a term appeared in document. So weight implies how relevant the term is for that particular document
- IDF is inverse document frequency calculated for incorporating measure that favors terms which occur in fewer documents. The fewer documents a term occurs in , the higher this weight.

www.ijmer.com Vol.2, Issue.3, May-June 2012 pp-945-948 ISSN: 2249-6645

- Thus this weight*IDF factor together will show a greater value if terms are important to document.

Id by terms matrix generation-

-Calculate weight of each term in all term list.
-Calculate Idf factor for each term in document under processing as:
Idf (i) = N/ni; where (N = no. of documents in repository, ni = No. of document in which term i occurred)
-Calculate W (i) = weight*Idf(i) for each term in the list
Now we will create (id X term) matrix as
-put W of term where there is match in all term list and doc id list
-put 0 where there is mismatch in all term list and doc id list

Query terms matrix generation-

it will be a single row matrix in which we-
When there is match in all term and query terms list
-put 10*idf if tree is not selected (query is not expanded and word matched in original query).
-put 5*idf if tree is selected and term matched is in the real query (query is expanded and word matched in original query).
-put 1*idf if tree is selected and term matched is in the expanded query (query is expanded and word is not matched in original query).
-Finally put 0 where there is mismatch in all term list and query terms list
This variation in first 3 weights is to incorporate the importance of a term if it is matched in a query and doesn't need to be expanded.

Result matrix and comparison-

Multiplication of above two matrix (id by term) and (query term) matrix will give a column matrix containing 0 in its row if there is a no word match in that particular file(having id same as row no.) and an integer value showing how many word have matched.
We can decide order of comparison by sorting results in descending order, thus file id having the most matches is at the top and lesser matches at their subsequent lower places.

IV. RESULT

We have tested our system on sample domain of computer science containing books of chapters and we are able to reduce the number of words to be searched in the file, thereby minimizing the search space. This effectively reduces the searching time as well. Reduction causes search spaces to be reduced more than 90% as our formula for selecting high frequency words finally used for index creation selects only such amount of words.
There are two type of comparisons are also performed which affects the search results. This are-
-Search using ontology or without it.
-Phrase based vs. term based search

In the first comparison using ontology recall increases more than 70% than without ontology if user enters words related to our domain in the query.
In the second comparison term based approach's results turned out be less relevant to the query in comparison to phrase based approach. For an example query operating system would fetch file system file as most relevant document while other would fetch operating system concept file as most relevant which is logically correct. This is because former approach uses both word operating and system as distinct term while later treats them as single. But recall in term based approach would be more as there are more terms to find out in the repository, also in case of single word query only term based approach would fetch a result.
Also second comparison may enjoy the benefits of using ontology in both case and hence can improve recall.
Although our searching method takes some time to index and then search the query but it reduces time of overall search in comparison to time required to search a document as a whole. By varying threshold of index creation we can vary the no. of words in document descriptive i.e. index table. We have also found that threshold value above a particular limit can eliminate some important words which is not desirable for our search. This limit depends upon the size of the documents we are using in our system.

V. CONCLUSION

In this paper we have described a technique which uses the concept of stemming so that a word can be searched using its root form and hence no need to be worried about query word's lexical forms. It also reduces search space by removing stopwords which are not helpful in search. . By varying threshold of index creation we can vary the no. of words in document descriptive i.e. index table. Our matrix multiplication approach finds out comparative results as which file are more relevant to the query and thus useful in ranked retrieval of documents. Use of ontology made a 70% recall for our system. Using phrase based approach with traditional term based approach we are able to increase relevancy between query and the result opted by user. Thus it shows an easy and fast approach to information retrieval.

VI. FUTURE WORK

We are currently in the process of attaching semantic meaning to this IR system both in query as well as documents.
We will try to have an Information retrieval system that adds word semantics to the classic word based indexing. Two of the main tasks of our system would be, the indexing and retrieval components, which will use a combined word based and sense-based approach. Main focus of our system would be a methodology for building semantic representations of open text, at word and collocation level. Also using ontology we will try to incorporate different relationships like is-a, has-a, part-of etc and make use of domain knowledge to make efficient search. Thus

incorporating semantic indexing approach with improved keyword based search approach overall efficiency of IR system can be improved.

VII. REFERENCES

[1] Using Structured Queries for Keyword Information Retrieval; Rajasekar Krishnamurthy Sriram Raghavan Shiva Kumar Vaithyanathan Huaiyu Zhu IBM Almaden Research Center, San Jose, CA 95120

[2] Toward industrial-strength keyword search systems over relational data; Baid, A.; Rae, I.; Anhai Doan; Naughton, J.F.; Comput. Sci. Dept., Univ. of Wisconsin, Madison, WI, USA.

[3] Callan, J. P. and W. Bruce Croft. An Evaluation of Query Processing Strategies using TIPSTER collection. In Proceedings of ACM SIGIR International Conference on Research and Development in Information Retrieval, 347-356, 1993.

[4] Rajashekar, T. B. and W. Bruce Croft. Combining Automatic and Manual Index Representations in Probabilistic Retrieval. Journal of the American Society for Information Science: 46(4), 272-283, 1995.

[5] Salton, G. Automatic Text processing. Addison-Wesley Publishing Company, Reading, MA: 1989.